UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

DISEÑO E IMPLEMENTACION DE SISTEMA DE DETECCIÓN AUTOMATICA DE PUBLICIDAD EN PRENSA ESCRITA

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELECTRICO

MAXIMILIANO RAMÍREZ MELLADO

PROFESOR GUÍA:
JOSÉ MANUEL SAAVEDRA RONDO

MIEMBROS DE LA COMISIÓN:
HÉCTOR AGUSTO ALEGRÍA
FRANCISCO GALDAMES GRUNBERG

SANTIAGO DE CHILE
2014

RESUMEN DE LA MEMORIA PARA OPTAR AL
TÍTULO DE INGENIERO CIVIL ELECTRICO POR:
MAXIMILIANO RAMÍREZ MELLADO
FECHA: JUNIO, 2014
PROF. GUÍA: JOSÉ MANUEL SAAVEDRA RONDO.

DISEÑO E IMPLEMENTACION DE SISTEMA DE DETECCIÓN AUTOMATICO DE PUBLICIDAD

La revisión sistemática de la prensa es una importante herramienta para el análisis de presencia de marcas y desarrollo de estrategias publicitarias, el monitoreo de estos medios tradicionalmente es realizado por operadores que manualmente extraen la información.

Este trabajo tiene como objetivo presentar un novedoso sistema de detección automática de publicidad sobre imágenes procedentes de diarios y revistas, dentro del marco del proyecto *IntelliMEDIA*, llevado a cabo por el *startup* chileno *CPDLabs*.

La metodología para detectar anuncios publicitarios se basa en un modelamiento específico de la estructura de la prensa escrita, permitiendo llegar a obtener una estimación del costo de los anuncios detectados. Para ello se separa el problema en 4 bloques: Preprocesamiento que separa el texto de las imágenes dentro de la página. Detección de logos que busca logos dentro de la página. Detección de publicidad que identifica el anuncio publicitario al que pertenece el logo y finalmente una etapa de tarificación que entrega una estimación del costo asociado al espacio publicitario.

El problema es abordado principalmente mediante tres estrategias: Primero la representación de la imagen en descriptores locales, que permite calzar características similares entre imágenes. Segundo, la estrategia de detección de objetos Viola-Jones, algoritmo de *machine learning* que genera un clasificador en base a un conjunto de imágenes de entrenamiento. La última estrategia es comparar histogramas de color permitiendo integrar información de color a la clasificación.

Para medir el desempeño de dichas estrategias se desarrolla un marco de evaluación, que consiste en una base de validación de 20.000 páginas de diario con 27 logos marcados, para así medir el desempeño de las distintas estrategias y configuraciones de parámetros, encontrar una solución eficaz para el problema y analizar las fortalezas y debilidades de los distintos métodos.

Los resultados demuestran que la solución es viable y es posible detectar logos mediante descriptores locales y *Viola-Jones*, logrando desempeños mayores al 90%. Por lo tanto IntelliMediA puede llegar a ser una manera eficaz y eficiente de extraer información publicitaria automáticamente de prensa escrita.

Tabla de contenido

Índice de ilustraciones

Índice de tablas.

Capítulo 1 - INTRODUCCIÓN

1.1 Motivación

Una gran cantidad de información es derivada desde la industria de la publicidad, generalmente el marketing en medios tradicionales involucra inversiones gigantescas. El monitorear esos esfuerzos monetarios es un problema complejo y no totalmente resuelto.

Por otro lado además del tema de la verificación de publicidad, la información publicitaria suele ser de sumo interés para quienes diseñan estrategias publicitarias, por ejemplo puede ser interesante para una empresa, conocer cuánto su competencia está invirtiendo en publicidad, que productos está promocionando, para así no ser superado mediáticamente por la competencia. También la empresa puede estar interesada en verificar que sus anuncios publicitarios sean correctamente emitidos por los medios.

Estos rubros, el *media monitoring* o *media clipping,* que se encuentran estrechamente relacionado con agencias de imagen, existen actualmente en forma de empresas que ofrecen servicios de monitoreo de publicidad sobre diarios, televisión o medios digitales, generalmente sus soluciones se encuentran basadas en operadores, que leen, miran y chequean anuncios publicitarios en los medios, proceso que es tedioso y poco efectivo, como se puede observar en la Ilustración 1.1.

Ilustración 1.1 Operadores de media *clipping*

IntelliMEDIA, el proyecto presentado en este trabajo, se presenta como una herramienta para solucionar el problema de manera automática, para el caso de la prensa escrita, entregando una solución eficiente y confiable, mientras que se reducen significativamente los costos del proceso al no depender directamente de operadores.

1.2 Antecedentes

1.2.1 Antecedentes generales

Ilustración 1.2 Chile País Desarrollado, la empresa involucrada. *IntelliMEDIA* el proyecto.

Este trabajo se encuentra enmarcado dentro del proyecto **INTELLIMEDIA**, parte de del *startup* chileno Chile País Desarrollado (Ilustración 1.2) dedicado al desarrollo de proyectos relacionados con procesamiento de señales e imágenes.

El proyecto **INTELLIMEDIA** fue parte del programa *Go-To-Market*[1] de *Corfo* en la edición 2013, programa enfocado a potenciar innovaciones tecnológicas con potencial comercial.

Se enmarca como un primer prototipo de un sistema de adquisición automática de metadatos publicitarios, principalmente como prueba de conceptos. Se plantea construir una solución, que usando estrategias del procesamiento de imágenes, detección de patrones y minería de datos, para el problema de la verificación de anuncios publicitarios de manera confiable y automática.

1.2.2 Antecedentes específicos

Para este trabajo se trabaja principalmente sobre *SimpleCV, OpenCV, VLFeat* y Matlab 2013.

Se trabaja usando logos obtenidos desde LHCV [1], imágenes de diarios digitales obtenidos desde el portal papel digital [2].

1.3 Objetivos

Objetivo General: Proponer una metodología capaz de detectar publicidad automáticamente en prensa escrita, mediante el uso de estrategias propias del procesamiento de imágenes, que a la vez sea confiable y eficiente permitiéndole ser un módulo funcional dentro de un sistema de reconocimiento de publicidad y finalmente llegar a ser aplicada en un sistema real de *media monitoring*[2].

[1] *Go-to-market* es un programa de corfo que financia talleres de innovación entregados por emprendedores exitosos.

[2] El monitoreo de medios es la actividad de monitorear la salida de medios escritos, online o televisión, puede ser realizado por una variedad de razones, incluyendo científicas comerciales y políticas. [28]

Objetivo Específicos:

- Describir de manera adecuada el problema, los requerimientos y los desafíos técnicos que requiere la solución completa.
- Estudiar distintos métodos para solucionar el problema de detección de logos.
- Diseñar una metodología ad-hoc al problema que lidie con las características particulares de la prensa escrita, incluido el tema de la tarificación de anuncios.
- Determinar los métodos más apropiados, mediante el diseño de pruebas y medidas de desempeño que sean acorde con el problema.
- Documentar el comportamiento de distintas estrategias y selecciones de parámetros y desarrollar un *benchmark* de distintas soluciones.
- Implementar finalmente un módulo de detección de logos y reconocimientos de publicidad con un desempeño aceptable.

1.4 Estructura de la memoria

Este documento está compuesto por las siguientes secciones.

Capítulo 1. Introducción: Se introduce el problema, y se presentan los alcances, objetivos y estructura de la memoria.

Capítulo 2. Revisión bibliográfica y contextualización: Se describe el problema y conceptos necesarios para comprender los requerimientos del sistema, luego se presentan estrategias usadas para diseñar la solución.

Capítulo 3. Diseño del sistema: Se presenta la solución, los requerimientos básicos de diseño y como se solucionan dichos requerimientos.

Capítulo 4. Desempeño de estrategias: Se realiza un *benchmark* para evaluar el desempeño de las estrategias para el problema específico de detección de logos, se presentan resultados y análisis de dichos resultados.

Capítulo 5. Conclusiones: Se presentan las implicancias de los resultados y su efecto sobre la factibilidad de la solución real para la detección de publicidad en prensa escrita.

Capítulo 2 – REVISION BIBLIOGRAFICA Y CONTEXTUALIZACION

En este capítulo se describen los conceptos generales básicos que dan marco al proyecto.

Temas a tratar

El objetivo de este trabajo es describir y desarrollar **un sistema automático de detección de publicidad en diarios y revistas**, para abordar el problema se debe contextualizar el tema explicando los siguientes temas:

- Los medios a analizar.
- El rubro publicitario.
- Aspectos técnicos relativos al procesamiento de imágenes.

Al ser un proyecto de integración tecnológica es necesario tratar desde aspectos técnicos, como los algoritmos de detección de objetos, hasta los temas específicos de esta aplicación, como puede ser describir precisamente que se entiende por publicidad dentro de prensa escrita.

2.1 Descripción del problema

Es necesario contextualizar el problema para poder abordarlo, para eso es necesario manejar ciertos temas relacionados con el rubro publicitario, y así generar una solución acorde al problema.

Se quiere diseñar una solución automática a la detección de publicidad en medios escritos, que además sea capaz de hacer una estimación de precios para los anuncios. Para ello se debe responder las siguientes preguntas:

- ¿Qué se quiere reconocer?
- ¿Dónde se quiere reconocer?
- ¿Cómo se quiere describir?

2.1.1 Descripción de medios

El objetivo es trabajar sobre prensa escrita tradicional sea esta:

- Periódicos
- Revistas

Básicamente un diario o una revista consta de un cuerpo, texto. Que viene acompañado por imágenes rectangulares como se puede observar en la Ilustración 2.1.

Ilustración 2.1 Partes relevantes de una página de diario

- **Cuerpo del diario**: Corresponde al contenido del diario, principalmente texto.
- **Publicidad**: Un anuncio publicitario, al cual se le quiere asignar un precio, tiene un tamaño definido (rectangular), y puede estar asociado a varios *logos/marcas*. Se asume que una publicidad relevante contiene al menos un logo reconocible.
- **Logos:** Es un elemento gráfico que identifica a una entidad o marca.
- **Texto**: El cuerpo del diario contiene texto e imágenes. El texto idealmente es filtrado.
- **Imágenes**: Imágenes pertenecientes al cuerpo de la noticia. Las imágenes que se encuentran dentro del cuerpo no pueden ser filtradas a priori, ya que podrían contener logos.
- **Encabezado**: El encabezado es el título que define la **sección** del diario de la actual página.

2.1.2 Descripción logos gráficos:

Un **logo** es una marca grafica o emblema usado comúnmente por empresas, organizaciones e individuos para ayudar a promover reconocimiento público de su entidad.

Los logos suelen ser **completamente gráficos** (Ilustración 2.2), compuestos por símbolos e iconos, o pueden ser **logotipos** (Ilustración 2.3), que están compuestos por el nombre de la organización

Ilustración 2.2 Logo Nike, completamente gráfico.

Ilustración 2.3 Logo la tercera, completamente tipográfico.

En teoría un logo podría ser cualquier imagen, aunque cabe destacar que los logos suelen tener las siguientes características:

- Suelen ser figuras **simples**, fáciles de recordar.
- Suelen tener **colores distintivos**.
- Suelen ser **simbólicos**.
- Suelen tener variadas representaciones graficas ligeramente diferentes.

Se considera que un logo puede ser **cualquier imagen**, para mantener la generalidad y no sesgarse a cierto tipo de logos al diseñar una estrategia de reconocimiento. Sin embargo, es requerido para que exista **presencia de marca** que la aparición del logo tenga al menos $\mathbf{1}cm^2$ [3]:

2.1.3 Tarificación de anuncios publicitarios

Uno de los objetivos fundamentales es obtener una estimación de los costos publicitarios en los que incurren las empresas que publicitan en prensa escrita.

Para realizar dicha estimación se debe estudiar cómo funciona la tarificación dentro de la industria de la publicidad. Usualmente una **empresa o agencia de publicidad**

[3] 1 cm mínimo en cualquiera de las dimensiones (w, h) $w > 1\ cm, h > 1cm$

compra un **espacio publicitario**, el cual tiene un costo definido y en teoría, transparente. En la práctica existen descuentos, alianzas y todo tipo de fallas de mercados que pueden hacer que un cliente pague más o menos que otro cliente por el mismo espacio publicitario, sin embargo el costo de los espacios publicitarios suele estar definido de antemano mediante un **tarifario,** documento que describe el cálculo del costo del anuncio.

El método más usado en la prensa escrita es la **tarificación modular**, que consiste en tarificar los **anuncios** en base a **módulos,** espacios publicitarios de tamaño definido. El **precio** de un anuncio queda completamente fijado por los siguientes parámetros:

- **Precio Modulo:** Depende del tamaño del módulo como se observa en la
- Ilustración 2.4.
- **Factor ubicación**: Considera sección, paridad…etc.
- **Factor día:** Considera el día en el cual se publica el anuncio.
- **Factor color:** Un cobro fijo por anuncios a color, que varía según el día de la semana y el tamaño del anuncio.

El valor del **anuncio** se puede obtener por la siguiente expresión:

$$(\boldsymbol{Precio\ Modulo\ x\ Factor\ ubicación\ x\ Factor\ día}) + \boldsymbol{Factor\ color} = \boldsymbol{Valor}$$

Ilustración 2.4: Precios modulares de tarificación, la tercera 2014.[4] [5]

[4] Tarifario la tercera 2014.

[5] Precios expresados en CLP.

En la Ilustración 2.4 se observa como los **módulos** definen un reticulado de 10x6 en la página del diario, por ejemplo si el ancho y alto de la página es (w, h), el módulo MD5x3 representa un rectángulo de ancho $\frac{3}{6}w$ y alto $\frac{5}{10}$h.

Para obtener el factor de ubicación es necesario **contextualizar** el anuncio, es decir, definir en qué cuerpo del diario se encuentra, si la página es par/impar, etc.…

El problema de **tarificar** presenta diversos desafíos técnicos, existen parámetros que parecen ser triviales de adquirir de manera automática, como por ejemplo el tamaño del módulo, el día en cual fue emitido el anuncio o si el anuncio es impreso en blanco y negro o en color, sin embargo existen parámetros que difícilmente se puedan identificar de manera automática, identificados como "interpretar contenido" en la Tabla 2.1

A continuación se especifican los parámetros que definen la **tarificación modular** de en el diario la tercera, en la Tabla 2.1.

Tabla 2.1: Parámetros de tarificación la tercera

		Factor	*Problema automatización.*
Factor ubicación	Generales	1	Detectar sección
	Crónica Par	2	Detectar sección
	Crónica Impar	2,5	Detectar sección
	Páginas Centrales	3	Interpretar numeración.
	Contraportada	3,5	Interpretar numeración.
	Solicitada	3	Muy difícil
	Impar hasta la 1	1,5	Interpretar numeración.
	Deportes Par	1,7	Detectar sección
	Deportes Impar	1,8	Detectar sección
	Espectáculos Par	1,7	Detectar sección
	Espectáculos Impar	1,8	Detectar sección
	Inserción	2	**Interpretar contenido**
	Legal y Balances	3	**Interpretar contenido**
	Proposición Política	1,1	**Interpretar contenido**
	Cine	1	Detectar sección
	Página Empresas B/N	1,4	**Interpretar contenido**
	Página Empresas (Color)	2	**Interpretar contenido**
	Martilleros	1	**Interpretar contenido**
Factor día	Lunes	1	Se obtiene directamente.
	Martes	1	Se obtiene directamente.
	Miércoles	1	Se obtiene directamente.
	Jueves	1	Se obtiene directamente.
	Viernes	1,15	Se obtiene directamente.
	Sábado	1,7	Se obtiene directamente.
	Domingo	1,6	Se obtiene directamente.

		Pequeño	Mediano	Grande
Factor color [6]	Lunes-Jueves	200.000	300.000	500.000
	Viernes	230.000	345.000	575.000
	Sábado	340.000	510.000	850.000
	Domingo	320.000	480.000	800.000

[6] Corresponde a un factor aditivo, los precios son expresados en CLP y dependen de si el anuncio es pequeño, mediano o grande, que corresponde a módulos rojos, amarillos y verdes respectivamente en la Ilustración 2.4.

2.2 Preliminares técnicos.

El objetivo de esta sección es entregar una síntesis de las estrategias utilizadas para solucionar el problema a través de la **visión computacional**, introducir conceptos que sean de interés para comprender los métodos utilizados y presentar un contexto técnico del problema.

Se realiza una revisión bibliográfica de temas relacionados con:

- Procesamiento de imágenes en general.
- Reconocimiento de objetos.
- Análisis de color.

2.2.1 Procesamiento de imágenes.

En esta sección se introducen conceptos relativos al procesamiento de imágenes, en particular aquellas implementadas en las etapas de **preprocesamiento** en el sistema, y el **procesamiento** del sistema, que cosiste principalmente en detección de objetos

2.2.1.1. Operaciones morfológicas.

La **erosión** [3] de la imagen binaria *A* por el elemento estructurante *B* está definida por:

$$A \ominus B = \{z \in E \mid B_z \subseteq A\}$$

$$B_z = \{b + z | b \in B\}, B_z: imagen\ B\ trasladad\ en\ z.$$

La erosión de *A* por *B* se puede entender como el lugar geométrico de los puntos alcanzados por el centro de *B* cuando *B* se mueve dentro de *A*. Por ejemplo, la erosión de un cuadrado de lado 10, centrado en el origen, por un disco de radio 2, también centrado en el origen, es un cuadrado de lado 6 centrado en el origen

La **dilatación** [3] es la operación opuesta a la erosión, se puede entender como un **crecimiento de pixeles**, es decir, la imagen binaria se expande a los pixeles adyacentes. Esto permite que aumente ciertos pixeles alrededor de la región y así poder incrementar dimensiones, lo cual ayuda a rellenar hoyos dentro de la región.

La dilatación de *A* por el elemento estructurante *B* se define por:

$$A \oplus B = \bigcup_{b \in B} A_b$$

La dilatación de *A* por *B* se puede entender como el lugar geométrico de los puntos cubiertos por *B* cuando el centro de *B* se mueve dentro de *A*.

2.2.1.2. Reconocimiento de regiones.

En el campo de la visión artificial, "***blob detection***" o **reconocimiento de regiones** se refiere a técnicas cuyo objetivo es detectar puntos o regiones más oscuras, o más claras de la imagen. Es importante ya que entrega información complementaria que no puede ser obtenida con otros enfoques como la detección de bordes/esquinas.

Generalmente se usan como una etapa previa al reconocimiento o detección de objetos, como es este caso.

Un enfoque práctico para detectar regiones más o menos brillantes es detectar **máximos locales en intensidad**, para luego asociar y extender una región a cada máximo local encontrado, sin embargo los extremos locales son muy sensibles al ruido.

Para enfrentar este problema **Lindeberg** [4], estudio la detección de máximos locales sobre varios **espacio-escala**[7] con el propósito de detectar regiones.

Algoritmo de detección de regiones de Lindeberg.

Para todos los pixeles de la imagen:

a) Si una pixel no tiene vecinos mayores es máximo local y por lo tanto es semilla de una región.
b) Si no, tiene por lo menos un vecino mayor, y ese vecino mayor es parte del fondo, no puede ser parte de una región y por lo tanto es fondo.
c) Si tiene varios vecinos mayores, pertenecientes a diferentes blobs es parte del fondo.
d) Si tiene varios vecinos mayores, que pertenecen todos al mismo blob, entonces pertenece a ese mismo blob.

[7] El estudio en espacio-escala es tratado en detalle en la sección 2.2.2

2.2.2 Reconocimiento y detección de objetos.

El **reconocimiento de patrones** es la ciencia que se ocupa de los procesos relacionados con el propósito de extraer información que establezca propiedades acerca de un objetos físicos o abstractos [5] , un esquema simple de un proceso de reconocimiento de patrones de **clasificación** se puede observar en la Ilustración 2.5.

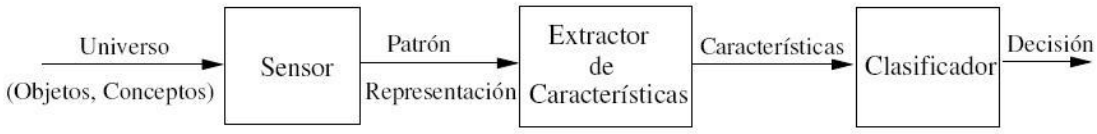


Ilustración 2.5 Reconocimiento de patrones

El proceso corresponde a extraer **características** que describan al objeto, que sean capaces de ser evaluados para clasificar al objeto, y tomar decisiones en base al sistema. Uno de los principales desafíos del reconocimiento de patrones es definir características acordes al problema que tengan propiedades discriminadoras, y definir **descriptores** que permitan hacer una representación de dichas características.

El **reconocimiento de objetos** es una aplicación particular de reconocimiento de patrones, en el campo de la visión computacional. Consiste en la tarea de identificar objetos dentro de una imagen o video, emulando la capacidad humana de reconocer objetos sin importar la **escala**, **rotación**, **posición** e incluso diferentes **perspectivas** o una **visión obstruida** del objeto. Es uno de los desafíos fundamentales de la visión computacional y diversos acercamientos han sido implementados a lo largo de décadas para solucionarlo.

El problema consiste en detectar si un objeto se encuentra dentro de una imagen, en este caso particular es necesario además definir la **pose del objeto,** que en una imagen queda definida como:

$$x, y, \sigma, \theta$$
$$x: Posicion\ x\ dentro\ de\ la\ imagen.$$
$$y: Posicion\ y\ dentro\ de\ la\ imagen.$$
$$\sigma: Angulo\ x\ del\ objeto\ en\ el\ eje\ de\ la\ imagen.$$
$$\theta: Escala\ del\ objeto, equivalente\ a\ eje\ z.$$

La **pose**, que tiene 6 dimensiones espaciales[8], queda representada en la imagen por **cuatro parámetros**, por lo que en la imagen se pierden 2 dimensiones rotacionales no triviales de recuperar, sin embargo existen metodologías que entregan robustez a variaciones en **Φ y φ**.

[8] La pose contiene 3 elementos de posición y 3 de orientación del objeto.($x, y, z, \sigma, \Phi, \varphi$).

La **detección de logos** sin embargo se realiza sobre objetos bidimensionales los cuales generalmente pueden ser correctamente identificados en la imagen usando 4 elementos de la pose.

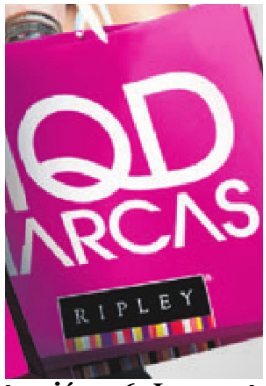


Ilustración 2.6: Logo rotado.

Ilustración 2.7: Logo con perspectiva distorsionada

En la Ilustración 2.6 se puede observar un logo rotado en la orientación σ, cuya pose queda totalmente descrita en la imagen, sin embargo en la Ilustración 2.7 el logo se encuentra rotado en una orientación distinta a la de la imagen, por lo que se considera una perspectiva distorsionada.

Además en el problema particular de la detección de logos, existen **distorsiones no lineales** debido a que las gráficas que representan a un logo pueden ser ligeramente distintas, como las gráficas de la Ilustración 2.8.

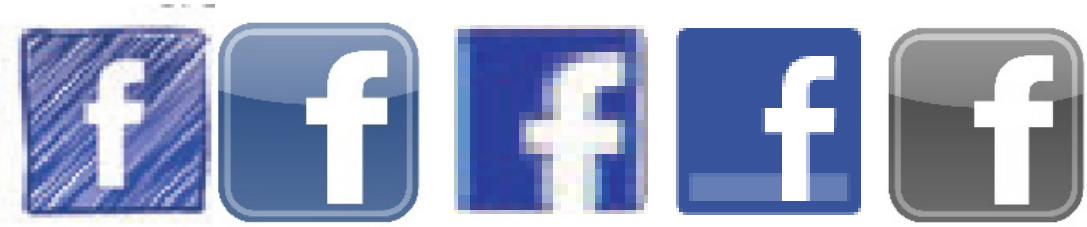


Ilustración 2.8 : Distintas instancias del logo Facebook.

Discusión:

Es interesante poner en el tapete si es realmente necesario usar algoritmos robustos de clasificación para un ambiente tan limpio, como lo es una página de diario, y no usar algo más rígido, como calce de *templates*.

Clasificadores robustos, como **SURF** y **Viola-Jones** son diseñados para tener un buen desempeño ante **rotaciones**, **oclusiones** y **cambios de perspectiva**, si bien estas distorsiones pueden estar presentes, la principal distorsión *intra-clase* entre logos en páginas de diario se debe a distintas representaciones graficas como en la Ilustración 2.8.

Una de las hipótesis interesantes en este trabajo es evaluar el desempeño de dichos clasificadores y enfrentarlos a *la creatividad de los diseñadores gráficos*. En la Ilustración 2.8 se puede observar distintas instancias del logo '*facebook*', que presenta distorsiones no lineales y sin embargo es muy fácil de identificar al ojo humano como un mismo objeto.

En este trabajo se abordan 3 enfoques para el reconocimiento de objetos.

- *Template based matching*.
- Descriptores globales.
- Descriptores locales.

En esta sección se describen distintos métodos usados para extracción de puntos de interés, descriptores, estrategias de calce, y otras metodologías usadas para detectar y reconocer logos y publicidades.

2.2.2.1. Métricas de desempeño

La salida del clasificador ante una muestra es la pose estimada del logo, por ejemplo si C es el clasificador que reconoce el **logo *l*** en una **página *P***:

$$C(P) = \begin{cases} (x, y, \sigma, \theta)_{estimado}, si\ se\ reconoce\ l\ en\ P \\ 0, si\ no \end{cases}$$

Si a priori se cuenta con el *groundtruth*[9], es posible evaluar el **desempeño** de un clasificador, asignando un valor de verdad a cada salida del clasificador, este valor puede ser:

$$VP: Verdadero\ positivo.$$
$$VN: Verdadero\ negativo$$
$$FP: Falso\ positivo.$$
$$FN: Falso\ negativo$$

$$V\big(C(Pagina)\big) = \begin{cases} VP: C(P) = (x, y, \sigma, \theta)_{estimado} = (x, y, \sigma, \theta)_{real} + \varepsilon \\ VN: C(P) = 0,\ \ l\ no\ se\ encuentra\ en\ P \\ FP: C(P) \neq 0,\ \ l\ no\ se\ encuentra\ en\ P \\ FN: C(P) = 0, l\ \ se\ encuentra\ en\ P \end{cases}$$

Para evaluar el **desempeño** del clasificador dentro de una base de datos es conveniente definir dos conceptos clave en clasificación:

$$Precision = \frac{VP}{VP + FP}$$

$$Recall = \frac{VP}{FN + VP} = \frac{VP}{Total\ de\ instancias}$$

[9] Dentro del campo de la evaluación de clasificadores, el *groundtruth* o base de verdad determina la exactitud en el proceso de clasificación a partir de un conjunto de entrenamiento clasificado manualmente.

La ***precision*** corresponde al porcentaje de aciertos dentro de todas las veces en que el clasificador detecta el objeto, mientras que el ***recall*** corresponde al porcentaje de aciertos entre todas las veces en que el objeto se encuentra en el *groundtruth*.

Por ejemplo un clasificador **poco exigente**, que reconoce el objeto en toda posición sin importar que se encuentre, tendría una exhaustividad de 1 y precisión cercana a 0, por otro lado un clasificador **demasiado exigente** tendría precisión cercana a 1 y exhaustividad cercana a 0, como ninguno de los casos se puede considerar un buen clasificador es necesario introducir una nueva medida, denominada **F-Score**:

$$F - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

F-Score corresponde a la media harmónica entre **precisión** y **recall** y permite medir el desempeño del clasificador sin estar fuertemente desbalanceado hacia clasificadores muy **exhaustivos**[10] o ***recall***, sino que privilegiando la harmonía entre los 2 indicadores.

El desempeño de un clasificador puede ser representado gráficamente mediante el uso de curvas características, que permiten visualizar la capacidad clasificadora del sistema ante distintos niveles de exigencia.

La curva ***Precision-Recall***, permite visualizar la precisión del sistema para distintas configuraciones. De esa manera se puede evaluar el desempeño sin fijar un nivel de exigencia particular, que depende de cada aplicación y del *tradeoff* existente entre la falsos positivos y falsos negativos en la aplicación.

[10] Exhaustividad es la característica de clasificadores con alto recall.

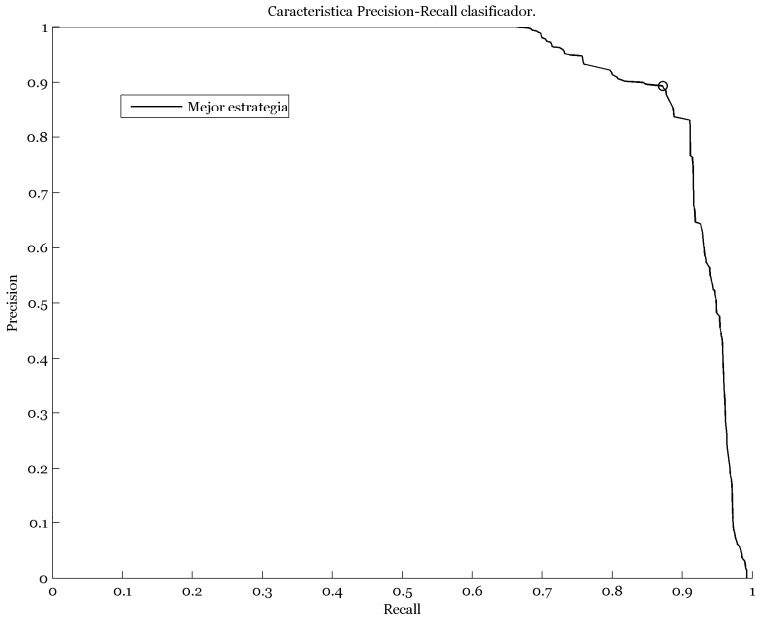


Ilustración 2.9 Curva precisión/recall.

La **curva precisión/*recall*** es por definición decreciente, ya que se ilustran las mejores configuraciones del sistema, es decir si una configuración logra un *recall* de 0.5 y precisión de 0.8 y existe una configuración que logra un *recall* de 0.6 y precisión de 0.85, la segunda configuración es sub-optima en precisión y *recall* y no es muestreada en la curva precisión/*recall*. Por otro lado si una configuración presenta un *recall* de 0.6 y precisión de 0.4 y otra un *recall* de 0.4 y precisión de 0.6, existe un *trade-off* entre ambas configuraciones y no es posible decir que una es mejor que la otra. Son estos los puntos que quedan graficados en la curva precisión/*recall*.

En la Ilustración 2.9 se observa una curva de *precisión/recall*. El círculo en la curva ilustra una configuración seudo-optima. Este punto corresponde a *recall* de 0.85 y precisión de 0.89, lo cual significa que existe una configuración de parámetros que hace que el clasificador logre dicho desempeño.

Por otro lado se introduce la **característica F-Score**, en ella se varía el umbral de clasificación en distintas configuraciones y para cada configuración se grafica precisión, *recall* y F-Score, en ella es posible visualizar la configuración seudo-optima que maximiza el F-Score, es decir el valor del umbral que maximiza F-Score.

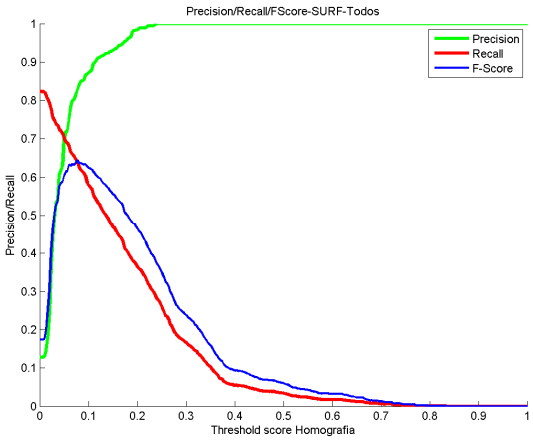


Ilustración 2.10 Ejemplo característica F-Score.

En la Ilustración 2.10 se observa como varían la precisión y el *recall* a medida que se varia el umbral de clasificación, cuando el umbral de clasificación es mayor, el clasificador es más restrictivo, logrando una precisión alta y un *recall* bajo. Por otro lado cuando el umbral es bajo, el recall es alto y la precisión es baja. Ambos parámetros se integran en su media harmónica, el F-Score, que tiene un máximo cerca de 0.1. Debido a que la precisión es creciente y el recall decreciente, la curva F-Score presenta un máximo y permite encontrar una configuración de umbral que entregue un buen desempeño.

2.2.2.2. Cálculo de homografía

En visión computacional se denomina **homografía** a una transformación proyectiva entre dos imágenes.

$$p'_b = H_{ab}\, p_a$$

$$p_a = \begin{pmatrix} x_a \\ y_a \\ 1 \end{pmatrix}, p'_b = w \begin{pmatrix} x_b \\ y_b \\ 1 \end{pmatrix}, H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}$$

La homografía puede ser calculada usando el algoritmo RANSAC, que busca una transformación que tenga un alto número de inliers, es decir puntos correctamente mapeados por la homografía, dado un umbral T_{dist}.

Algoritmo RANSAC:
Dados N_m candidatos a calce:

1. Se inicializa el número de estimaciones $N = 1500$, el umbral T_{dist} , el valor $MAX_{inlier} = -1$, MIN_{std}=100000
2. Para cada $1 = 1{:}\,N$
   a. Se eligen cuatro candidatos a calces al azar.
   b. Se verifica si los puntos son colineales, si es así, se repite a.
   c. Se computa una homografía H_{actual}, usando los 4 calces.
   d. Para cada correspondencia se calcula el error de estimación de la homografía H_{actual}, usando todos los puntos.
   e. Se computa la desviación estándar del error de estimación de la homografía std_{actual}.
   f. Se cuenta el número de calces m que cumplen $d_i < T_{dist}$
   g. Si $m > MAX_{inlier}$ && $std_{actual} < MIN_{std}$, H_{actual} pasa a ser el candidato a homografía.

En [6] Torr presenta una variante a RANSAC llamada **MLSAC**, una generalización de RANSAC que introduce el concepto de **verosimilitud** [11], generando medidas de verosimilitud en vez de basar el cálculo de homografía en maximizar el número de *inliers*.

Es de interés notar que para lograr mayor precisión en la clasificación es posible fijar condiciones sobre la matriz de homografía, como se muestra en la siguiente tabla.

Tabla 2.2 Condiciones sobre matriz de homografía

Transformación afín	$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{pmatrix}$
Transformación afín sin rotación	$H = \begin{pmatrix} h_{11} & 0 & h_{13} \\ 0 & h_{22} & h_{23} \\ 0 & 0 & 1 \end{pmatrix}$
Traslación pura	$H = \begin{pmatrix} 1 & 0 & h_{13} \\ 0 & 1 & h_{23} \\ 0 & 0 & 1 \end{pmatrix}$
Traslación, escalamiento en k y rotación en Θ	$H = \begin{pmatrix} \mathrm{k}\cos(\Theta) & -k\sin(\Theta) & h_{13} \\ k\sin(\Theta) & \mathrm{k}\cos(\Theta) & h_{23} \\ 0 & 0 & 1 \end{pmatrix}$
Etc...	

[11] Verosimilitud o likelihood es la idea de que tan posible es un evento.

La matriz de homografía es de interés ya que permite representar la posición de la imagen *template*, que contiene al objeto que se quiere reconocer, dentro de la imagen sample, y por lo tanto representar la posición del objeto.

2.2.2.3. Template-based matching

Template based matching es una técnica en procesamiento de imágenes para encontrar la parte de una imagen que calza con una imagen *template*, comparando niveles de intensidad. Consiste básicamente en encontrar la **traslación** donde la **diferencia** entre las dos imágenes es **mínima**, asignándole así una posición a la imagen template en la imagen sample.

La **métrica** del calce es la diferencia en valor absoluto entre la imagen template trasladada $T\left(I_{i,j}^{T}\right)$ y la imagen sample I^{S}.

$$d(x,y) = \sum_{i=1}^{n} \left| T\left(I_{i,j}^{T}\right) - I^{S} \right|$$

El calce puede ser encontrado por **fuerza bruta** iterando sobre las posibles traslaciones.

Sin embargo este método no es invariante a escala, rotación ni distorsiones, es muy poco robusto y si se quiere lograr **invariancia** ante escala y rotación se debe usar distintos ***templates***, lo cual puede llegar a ser muy costoso computacionalmente.

2.2.2.4. Framework de detección de Viola Jones.

Viola-Jones es una estrategia de *machine learning*[12] que es capaz de conseguir altos índices de detección eficientemente. Hace uso de la **imagen integral**, una construcción que permite calcular rápidamente sumas de áreas, para lograr una alta eficiencia. Está basado en el algoritmo de aprendizaje *AdaBoost* [7], que selecciona un número de **características visuales** críticas sobre un set de características y permite crear clasificadores extremadamente eficientes, también incorpora un método para integrar clasificadores complejos **en cascada**, que permiten descartar rápidamente regiones de que no son de interés en la imagen.

Esta metodología es de particular utilidad en el campo de reconocimiento facial, sin embargo, la herramienta está planteada en el contexto general del reconocimiento de objetos y se puede entrenar con cualquier **conjunto de *templates***.

[12] ***Machine learning*** o aprendizaje de máquinas es una rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender.

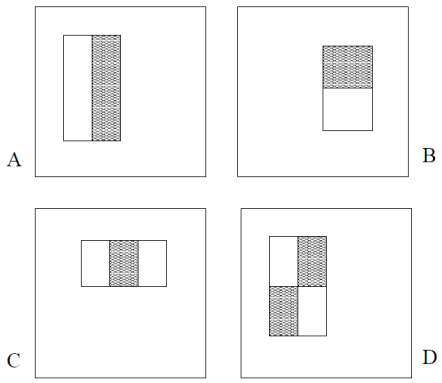


Ilustración 2.11 : Cuatro características rectangulares obtenidas de una imagen.

Características:

El detector Viola-Jones se basa en el valor de **características simples** que son similares a **filtros Haar**[13]en la Ilustración 2.11. Se usan 3 tipos de características simples, la figura A presenta 2 rectángulos, donde el valor de la característica es la suma del rectángulo negro restada con el rectángulo blanco, análogamente se describen características sobre 3 rectángulos (figura C) y 4 rectángulos (figura D), considerando un descriptor de 24x24, existe un total de 180,000 posibles características, por lo que es conveniente usar *AdaBoost* para seleccionar aquellas características relevantes.

Entrenamiento

Se define un clasificador débil $\boldsymbol{h_j(x)}$ aquel que considera una sola característica $f_j(x)$.

$$h_j(x) = \begin{cases} 1 \; si \; p_j f_j(x) < p_j \Theta_j \\ 0 \; si \; no \end{cases}$$

$$f_j(x): Caracteristica \; simple \; de \; la \; imagen.$$

$$p_j: Paridad, permite \; invertir \; la \; desigualdad.$$

$$\Theta_j: Umbral \; del \; clasigicador$$

Como cada característica es muy simple, en la práctica ningún clasificador débil logra un desempeño aceptable.

[13] Filtros Haar es una familia de funciones presentada en [33].

El algoritmo **AdaBoost**[14] permite integrar estos clasificadores débiles.

***AdaBoost*:**

- Dadas imagenes ejemplo (I_1, y_1), ... (I_n, y_n), donde:

$$y_i = \begin{cases} 1\ ejemplo\ positivo \\ 0\ ejemplo\ negativo \end{cases}$$

- Se inicializan los pesos $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ para y_i =0,1 respectivamente, donde m es el número de positivos y l el número de negativos.
- Para t=1,....,T. T: número de clasificadores débiles.
  - Se normalizan los pesos.

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$

    Para que la w_t sea una distribución de probabilidad.

  - Para cada característica, j, se entrena un clasificador débil que está restringido a usar una sola característica, el error es por el peso.

$$\varepsilon_j = \sum_i w_i \, |h_j(x_i) - y_i| \, m$$

  - Se elige el clasificador h_t con menor error ε_{kt}.
  - Se actualizan los pesos:

$$w_{t+1,i} = w_{t+1,i} \, \beta_t^{1-e_i}$$

$$e_i = \begin{cases} 0\ si\ la\ clasificacion\ es\ correcta \\ 1\ si\ no. \end{cases}$$

$$\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$$

- El clasificador fuerte final es:

$$h(x) = \begin{cases} 1\ si \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0\ si\ no \end{cases}$$

$$donde\ \alpha_t = \log \frac{1}{\beta_t}$$

[14] La formulación de *AdaBoost* expuesta en [23]

En cada ronda de ***AdaBoost*** se selecciona una característica de un potencial de 180.000, la cual termina generando un conjunto de T características que componen al clasificador.

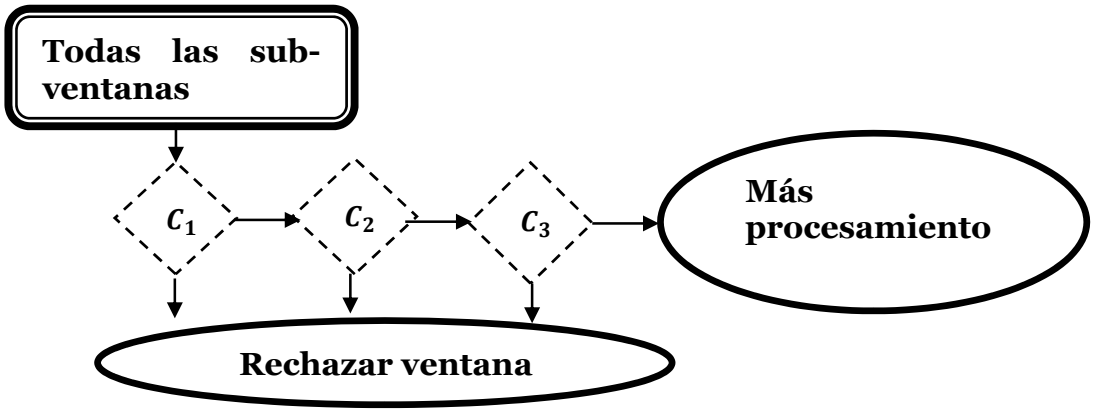


Ilustración 2.12: Detectores en cascada

Se plantea también usar una **cascada** de filtros entrenados por *AdaBoost*, como se observa en la Ilustración 2.12 , usar cascadas de clasificadores es extremadamente importante en aplicaciones donde el costo computacional es limitante, ya que se puede descartar una ventana rápidamente, sin incurrir el costos computacional de calcular todas las características.

La imagen integral:

El uso de la **imagen integral** es clave para el *framework Viola-Jones*, como también para otras aplicaciones como SURF. La imagen integral en el punto (x,y) consiste en la suma de intensidades de todos los puntos (i, j) donde $i < x\ y\ j < y$.

$$Imagen\ integral: I_{\Sigma}(x, y) = \sum_{i=1}^{i<x} \sum_{j=1}^{j<y} I(i, j)$$

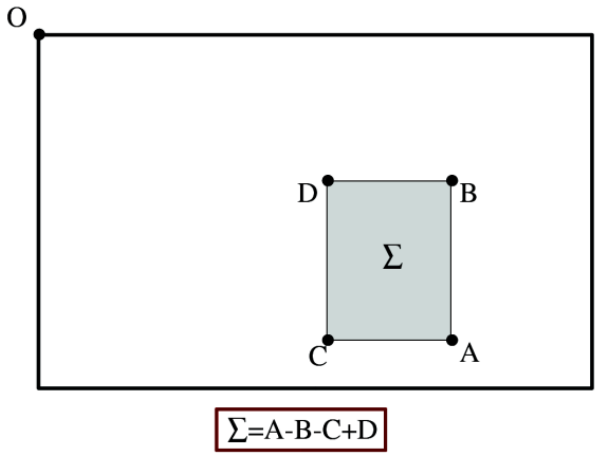


Ilustración 2.13 Resta de imagen integral para obtener áreas rectangulares.[15]

La Ilustración 2.13 muestra gráficamente cómo, una vez calculada la imagen integral, es posible calcular cualquier área rectangular mediante tres sumas, independiente del tamaño del rectángulo. Así todos los cálculos de áreas de las características simples pueden ser calculados como la suma y resta de 4 puntos en la imagen integral.

2.2.2.5. Descriptores locales.

Existen diversas estrategias basadas en descriptores locales, entre ellas SURF, SIFT, FREAK, BRISK. Que son tratadas en este capítulo.

Las estrategias de detección de objetos basadas en descriptores locales siguen la metodología planteada por Lowe en SIFT [8], Consiste básicamente en:

- **Localización de puntos de interés:** Calcular **puntos de interés** o ***features*** en la imagen, los cuales tienen asignado posición, orientación y un factor de escala.
- **Cálculo de descriptores:** Para dichos puntos de interés se calcula un **descriptor local** invariante a escala, que describe la región que rodea al punto de interés.
- **Calce de descriptores:** Se encuentran similitudes entre los **descriptores** de una imagen *sample* y una imagen *template*. Dichas similitudes generan **calces**.
- **Verificación de calces:** Para cada **calce** se hace una verificación geométrica entre los puntos de interés que los generan. Esto se logra haciendo el cálculo de una homografía que haga calzar las coordenadas de los puntos de interés. Así, se obtienen **calces validados**. Estos calces permiten hacer correspondencia entre imágenes y detectar la presencia del objeto de la imagen template en la imagen sample.

[15] Ilustración obtenida de [9].

Si bien las estrategias que se trabajan a continuación varían en las estrategias de localización de puntos de interés, cálculo de puntos de interés, calce de puntos de interés y verificación de calces, todas se basan en **SIFT**, explicado a continuación en detalle.

2.2.2.6. Scale-invariant feature transform (SIFT).

SIFT [8] es una metodología que permite identificar puntos de interés en una imagen, y sobre cada punto de interés, extraer un conjunto de vectores invariantes a escala y rotación, llamados **descriptores locales**.

Cada **descriptor** tiene asociado la posición, orientación y escala del punto de interés detectado, de esa manera los descriptores SIFT son usados para determinar características similares entre distintas imágenes. Se puede hacer reconocimiento de objetos haciendo coincidir los **descriptores** de una imagen *sample*, con los descriptores previamente calculados de una *imagen template*, que se quiere reconocer. Para ello Lowe plantea usar un algoritmo de **vecinos cercanos**, encontrar la correspondencia entre los descriptores de dos imágenes, y luego una etapa que permite identificar si las correspondencias presentan posiciones, rotaciones y escalas **coherentes**. Finalmente se puede encontrar una transformación de coordenadas que puede relacionar ambas imágenes y así encontrar la ubicación de la imagen *template* dentro de la imagen *sample*.

De esa manera SIFT es una robusta estrategia para identificar objetos escalados, rotados o con oclusiones.

Algoritmo

La detección **de puntos de interés** se hace usando espacio de escalas (*Scale-space Detection*) Este enfoque trata de identificar posibles puntos de interés usando la imagen original y una cascada de filtros, donde cada filtro representa una distinta escala. Así se pueden encontrar puntos de interés en distintas escalas y por ende lograr la invariabilidad a escala en el descriptor.

Se define el **espacio-escala** (*Scale Space*) como una cascada de funciones $L(x, y, \sigma)$, que son la convolución entre la imagen original y una gaussiana, de varianza σ, de esa manera el efecto del filtro es difuminar la imagen, perdiendo resolución.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

Donde * es el operador convolución y la gaussiana es:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Luego para localizar los puntos de interés se implementa la función **diferencia de gaussiana** (DoG):

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

Donde **k** es un factor multiplicativo que representa el cambio de escala.

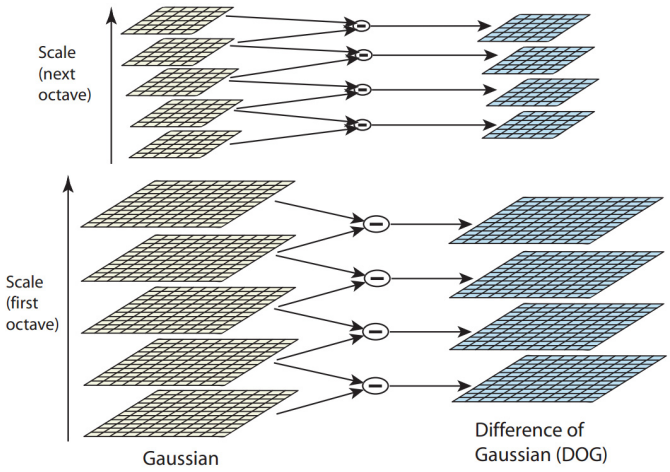


Ilustración 2.14 Diferencia de Gaussianas, DOG[16]

En la Ilustración 2.14 se puede observar la construcción de la **pirámide DoG**. La imagen inicial se convoluciona con el filtro Gaussiano con σ = 2, luego, de la diferencia entre la imagen en dos distintas escalas se obtienen los pisos de la pirámide. Después la imagen es re muestreada una octava más gruesa[17] y se siguen obteniendo diferencias de gaussianas.

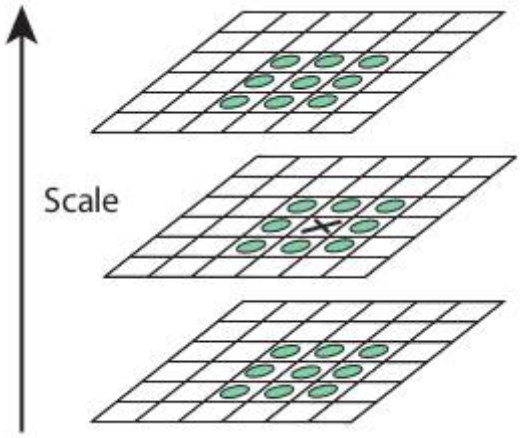


Ilustración 2.15 Extremos locales en espacio y escala

Una vez calculada la pirámide, se buscan los extremos locales en **espacio y escala**, comparando cada punto con sus ocho vecinos en las escala inmediata, al igual con los 18 vecinos de las escalas anterior y posterior con distinto σ. En la Ilustración 2.15 se puede observar el extremo local y los vecinos.

[16] Imágenes obtenidas de [8].

[17] Se obtiene una escala más gruesa re-muestreando la imagen a una imagen con la mitad de resolución.

Localización del punto de interés: Para localizar el punto de iteres, se realiza una interpolación entre cada punto y los vecinos en escalas superiores e inferiores. Luego se interpola el máximo usando una parábola para localizar del punto.

Además se definen dos umbrales para rechazar aquellos puntos de interés considerados como no relevantes:

- Se eliminan los puntos de interés con un valor menor a 0.03 veces el máximo en DoG.
- Se eliminan los puntos de interés que tengan vecinos mayores a 0.95 veces el máximo en un radio de 5 pixeles, comparando con la misma escala.

Orientación del punto de interés: Para lograr invariancia a rotación se asigna una dirección al punto de interés, basada en el gradiente local en la imagen.

En la imagen suavizada L(x, y, σ), se calcula:

$$m(x,y) = \sqrt{\big(L(x+1,y) - L(x+1,y)\big)^2 + \big(L(x,y+1) - L(x,y-1)\big)^2}$$

$$\theta(x,y) = tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{(x+1,y) - L(x-1,y)}\right)$$

Para asignar orientación a la región, se calcula un **histograma de orientaciones** formado con 36 bines[18] que cubren 360 grados. Cada muestra tiene un peso que se pondera por la magnitud del gradiente, y por la distancia al centro de la región del punto de interés, ponderada con una ventana gaussiana.

Luego el bin con más votación se considera para asignar orientación al punto de interés, además, si existen celdas con 80% del valor del máximo, estos sirven como origen de un nuevo punto de interés con distinta orientación θ.

De esa manera quedan localizados los puntos de interés $\boldsymbol{f_i}$ en la imagen.

$$f_i^I \in F, F^I: Puntos\ de\ interes\ de\ la\ imagen\ I.$$

$$f_i^I = (x, y, \sigma, \theta, m)$$
$$x: Posicion\ x.$$
$$y: Posicion\ y.$$
$$\sigma: factor\ de\ escala.$$
$$\theta: Orientacion.$$
$$m: Valor\ del\ gradiente.$$

[18] Un bin es una ventana sobre la cual se mapean datos, generando un histograma.

Descriptor SIFT:

La etapa previa muestra como asignar **localización**, **orientación** y **escala** a los puntos de interés, estos representan una región local de la imagen. Para poder encontrar coherencias entre puntos de interés es necesario caracterizar dichas regiones con un **descriptor local** que sea altamente distintivo e invariante a variaciones, tales como cambios de iluminación o perspectiva.

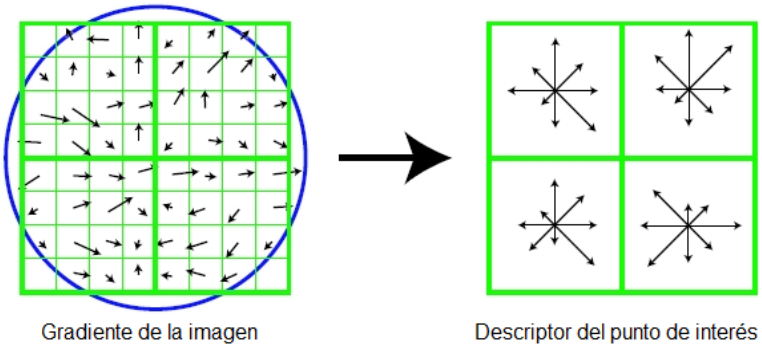


Ilustración 2.16 Descriptor SIFT simplificado 2x2

En la Ilustración 2.16 es posible observar el esquema para generar el descriptor de la imagen. Se calcula la magnitud del gradiente en el parche que rodea al punto de interés, para asignar un peso a la magnitud en cada punto se usa una ventana gaussiana, representada en la imagen como una ventana circular azul, dando así menos énfasis a puntos más lejanos al punto de interés. Estos puntos son acumulados en un histograma de orientación, que se resume en 16 sub-regiones, para los cuales se obtiene el histograma de orientaciones en cada dirección representados en la Ilustración 2.16 como vectores en las 8 direcciones principales del gradiente.

De esa manera se calcula un descriptor de **128 bytes** para cada punto, que es normalizado para lograr invariancia a la iluminación, de esa manera, el descriptor hace énfasis en la distribución de la orientación de los gradientes, como una característica invariante a escala, orientación y distorsiones.

Los puntos de interés de la imagen I quedan caracterizados como:

$$d_i{}^I \in D, D^I: Puntos\ de\ interes\ de\ la\ imagen\ I.$$

$$d_i{}^I = \begin{pmatrix} d_i{}^I(1) \\ \vdots \\ d_i{}^I(128) \end{pmatrix}, vector\ de\ 128\ bytes$$

Estos puntos de interés quedan representados en la Ilustración 2.17, donde el tamaño de los círculos representa la escala del punto de interés.

Ilustración 2.17 Puntos de interés en representación gráfica.[19]

Calces de puntos de interés: Cuando se quiere hacer calzar una imagen *sample* y *template* se realiza calce entre los puntos de interés. El mejor candidato de calce para cada punto de interés se encuentra calculando los vecinos más cercanos, que es definido como el punto de interés de la imagen *template* cuyo descriptor tiene la mínima distancia euclidiana con el descriptor de la imagen *sample*.

Además Lowe [8], propone evaluar que el radio entre las distancias del primer y segundo vecino más cercano sea 0.8, para hacer énfasis en la singularidad del calce.

Se define el conjunto de calce entre dos imágenes T y S como:

$$M = (k, q)\ tq\ \begin{cases} dist\left(d_k{}^T, d_q{}^S\right) \leq Threshold. \\ dist\left(d_k{}^T, d_q{}^S\right) < 0.8\left(\ dist\left(d_k{}^T, d_j{}^S\right)\ \right), \qquad \forall\ d_j{}^S \in D^S, d_j{}^S \neq d_k{}^T \end{cases}$$

$$d_k{}^T \in D^T, descriptor\ de\ la\ imagen\ template.$$
$$d_q{}^S \in D^S, descriptor\ de\ la\ imagen\ sample.$$

Validación de calces: Cuando se buscan los **calces**, simplemente se considera la distancia entre los vectores descriptores, sin darle importancia a la coherencia geométrica de dichos calces, muchos de los calces son fortuitos, y debe ser identificada una coherencia geométrica entre los puntos de interés de una imagen y la otra, es decir calcular la **homografía**. La diferencia entre calces no validados y calces validados se puede ver en la Ilustración 2.18.

[19] Obtenido de [30].

Ilustración 2.18 : Calces vs calces validados

La homografía debe cumplir que:

$$\forall (k,q) \in M^V, \qquad \left| H\left(f_q{}^T\right) - f_k{}^S \right| \leq Threshold$$
$$M^V \subseteq M,\ M^V\text{: Conjunto de inliers, calces validados.}$$

De esa manera se define el score del calce como:

$$score = \frac{\| M^V \|}{\| F^T \|}$$
,

Ósea el porcentaje de puntos de interés calzados en el total de puntos de interés de la imagen *template*, la posición estimada del objeto en la imagen *sample* se puede obtener calculando un ***bounding box***[20] que encierra el calce:

$$BBox_e = \begin{pmatrix} x1_e, y1_e \\ x2_e, y2_e \end{pmatrix} = \begin{pmatrix} H(0,0) \\ H(w,h) \end{pmatrix}$$
$$w, h\text{: Ancho y alto de el template.}$$

Además es posible hacer descarte de calces mediante la interpretación de la **matriz de homografía**, por ejemplo si los requerimientos del clasificador no incluyen invariancia a escala, se pueden descartar calces mediante la interpretación de la matriz de homografía, reduciendo así el índice de falsos positivos y por lo tanto aumentando la precisión del clasificador.

La metodología descrita en este apartado: detectar puntos de interés, describir regiones contiguas a dichos puntos y luego calzar dichos puntos es usada por diversas estrategias como SURF [9], BRIEF [10], BRISK [11] , ORB [12], FREAK [13], y ha sido demostrada como una de las mejores estrategias para reconocer objetos.

[20] *Bounding* box es un rectángulo que contiene a un objeto dentro de una imagen.

2.2.2.7. *Speeded-Up Robust Features* (SURF)

SURF es una metodología que permite detectar puntos de interés y sobre ellos calcular un descriptor invariante a escala y rotación [9] , tiene la particularidad de presentar alta repetitividad[21] ante variaciones y por lo tanto es robusto, a la vez que es más eficiente que **SIFT** en cuanto a costos computacionales.

Detección de puntos de interés

Para detectar puntos de interés se plantea un esquema basado en el cálculo de la **imagen integral**, que permite el cálculo rápido de la convolución en regiones cuadradas, que permite el cálculo de la **matriz hessiana**.

Para un punto (x,y) la matriz hessiana de escala σ se define como**:**

$$H(x,y,\sigma) = \begin{bmatrix} L_{xx}(x,y,\sigma) & L_{xy}(x,y,\sigma) \\ L_{yx}(x,y,\sigma) & L_{yy}(x,y,\sigma) \end{bmatrix}$$

$$L_{xy}(x,y,\sigma) = \frac{\partial^2 G(x,y,\sigma)}{\partial x \partial y} * I(x,y)$$

Estos filtros gaussianos son óptimos para el análisis en escala-espacio, pero de todas maneras deben ser discretizados, por lo que se plantean aproximaciones de esos filtros $D_{xx}, D_{xy}, D_{yx}, D_{yy}$, que permiten hacer uso del cálculo de áreas por medio de la imagen integral, como se puede observar en la Ilustración 2.19, donde a la izquierda se observan los filtros L_{yy} y L_{xy} y a la derecha sus aproximaciones D_{yy} y D_{xy}.

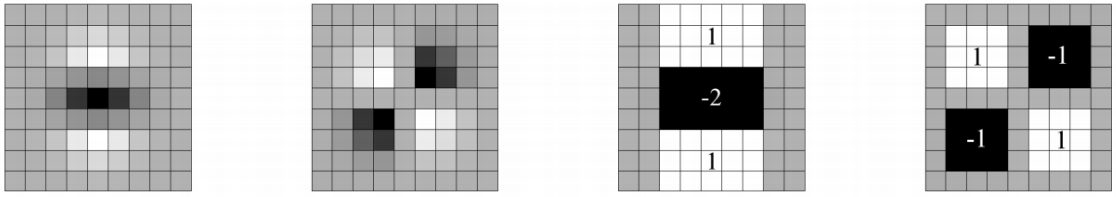


Ilustración 2.19: Los filtros gaussianos derivativos discretosD_{xx}, D_{xy} y sus aproximaciones.[22]

Una vez calculados $D_{xx}, D_{xy}, D_{xy}, D_{yy}$, a partir de la imagen y los filtros aproximados, se calcula el determinante aproximado de la matriz Hessiana, cuyos máximos locales son considerados puntos de interés.

$$det(H_{approx}) = D_{xx}D_{xy} - 0.9{D_{xy}}^2$$

[21] Repetitividad es la característica de descriptores de ser invariante a variaciones.

[22] Imágenes obtenidas de [9].

El factor 0.9 aparece como corrección al usar filtros rectangulares aproximados.

Usando filtros de distinto tamaño la metodologia SURF es capaz de hacer representacion de la imagen en espacio-escala, donde filtros de mayor tamano representan escalas mas gruesas, usando filtros de tamano 15x15, 21x21, 27x27,... 51x51, 99x99, 147x147. Como se puede observar en la Ilustración 2.20.

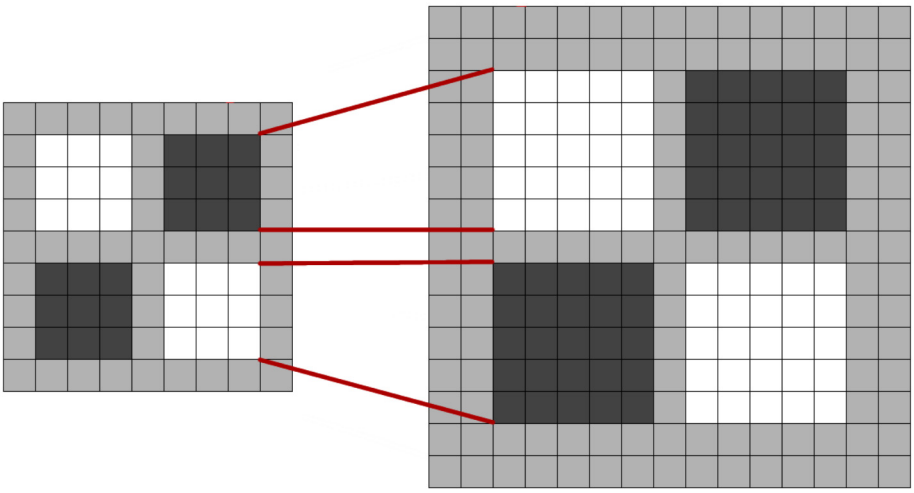


Ilustración 2.20: Filtros aproximados en distintas representaciones de escala, 9x9 y 15x15.

Para localizar los puntos de interés se buscan los máximos locales de la matriz hessiana, en escala y espacio y luego se interpola para obtener una ubicación precisa del punto de interés.

Descriptor SURF

El descriptor SURF se basa en la respuesta filtros Haar en el área del punto de interés, que permiten hacer una representación de la distribución de intensidad.

Ilustración 2.21: *Haar wavelets* de primer orden en x e y.

El descriptor se construye de manera análoga al descriptor SIFT, solo que en vez de calcular el gradiente en la vecindad del punto de interés, se calcula la respuesta a los **Haar wavelets** mencionados en la Ilustración 2.21, con lo que se tiene una reducción considerable del tiempo de cálculo.

Se asigna **orientación** usando una vecindad circular alrededor del punto de interés, sobre la cual se calcula la respuesta de los wavelets, ponderada por una gaussiana

centrada en el punto de interés. Luego se busca, dentro de la vecindad circular, la ventana de tamaño $\frac{\pi}{3}$, donde la suma de la respuesta es máxima. Dicha ventana se puede observar en la Ilustración 2.22.

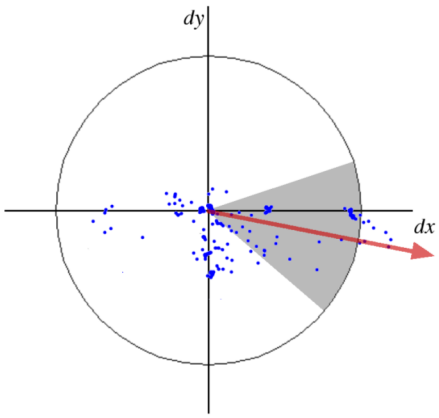


Ilustración 2.22: Asignación de orientación basada en ventana $\frac{\pi}{3}$.

Para construir el descriptor se considera una ventana cuadrada de tamaño 20s, donde s es el factor de escala del punto de interés, sobre los cuales se calcula la respuesta a los filtros *Haar* de primer grado, ponderada por una gaussiana centrada en el punto de interés, como queda graficado en la Ilustración 2.23

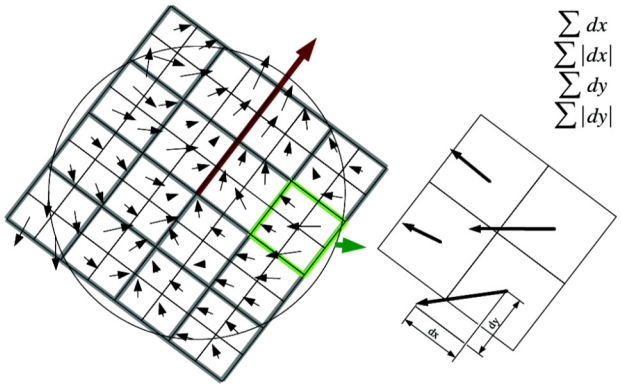


Ilustración 2.23: Construcción del descriptor SURF, para cada área de 2x2 (en verde), se calculan los índices $\sum dx, \sum dx, \sum dx, \sum dx$, que en el total de 16 áreas constituyen un vector de tamaño 16x4=64.

La región cuadrada de tamaño 20*s* es dividida en 4x4 subregiones[23], preservando así la información espacial, para cada una de esas subregiones se calcula la respuesta de los filtros *Haar*, y se obtienen las respuestas horizontales d_x y las verticales d_y. De estas respuestas se obtiene la suma y suma en valor absoluto generando el vector de intensidades $v = (\sum d_x, \sum d_y, \sum|d_x|, \sum|d_y|)$, de las 16 regiones se obtiene un vector de **largo 64** en total.

[23] s corresponde a la escala en la cual se encuentra el punto de interés.

Los puntos de interés de la imagen I quedan caracterizados como:

$$d_i{}^I \in D, D^I: Descriptores\ de\ la\ imagen\ I.$$

$$d_i{}^I = \begin{pmatrix} d_i{}^I(1) \\ \vdots \\ d_i{}^I(64) \end{pmatrix}, vector\ de\ 64\ bytes.$$

Calce de puntos de interés:

La etapa de calces de puntos de interés es similar a SIFT, la etapa de validación también, con la única diferencia de que el descriptor integra un elemento que representa el **signo del gradiente**, que permite descartar calces tempranamente.

2.2.2.8. *Fast retina keypoints*(FREAK).

FREAK [13] es otra estrategia para obtener descriptores sobre puntos de interés, que aparece como un método práctico de hacer calce de descriptores locales eficientemente, con una particular utilidad en aplicaciones móviles, o aplicaciones en tiempo real, donde las consideraciones de memoria y eficiencia son limitantes.

Freak plantea un descriptor novedoso, inspirado en el funcionamiento del sistema visual humano, en particular la retina como se puede observar en la Ilustración 2.24. Se computa una cascada de vectores binarios comparando las intensidades de imágenes sobre un patrón de muestreo retinal[24]. Experimentos realizados en [13] dicen que FREAK presenta una significativa mejora en cuanto a la carga computacional, y a la vez es más robusto que otros descriptores como SIFT, SURF y BRISK[25].

Una ventaja de usar vectores binarios es que el calce se puede hacer usando distancia de hamming, lo cual reduce el orden de complejidad de la tarea de calzar los puntos de interés.

[24] La retina es un tejido sensible a la luz situado en la superficie interior del ojo. Es similar a una tela donde se proyectan las imágenes.

[25] BRISK, *Binary Robust Invariant Scalable Keypoints*, es otro descriptor local binario que precede a FREAK.

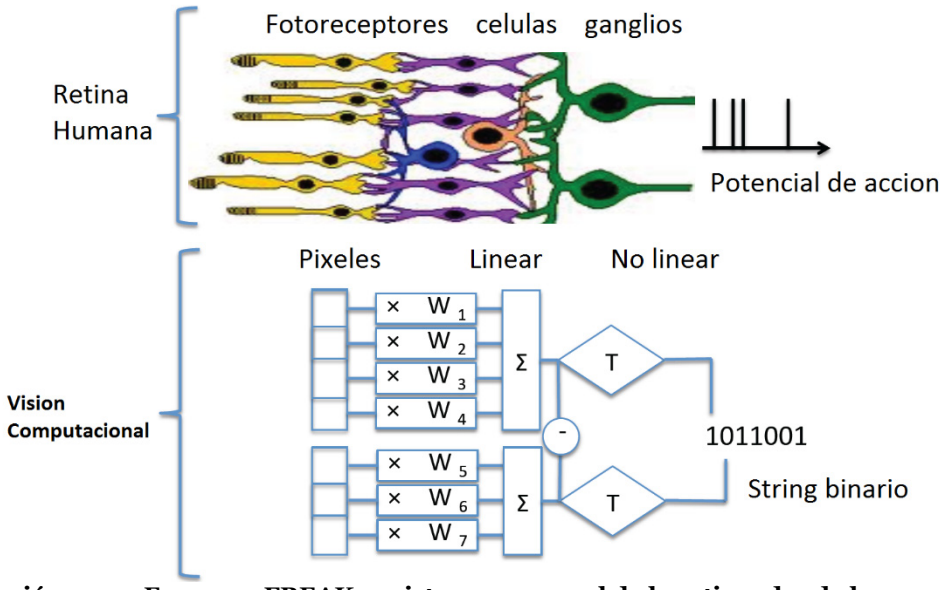
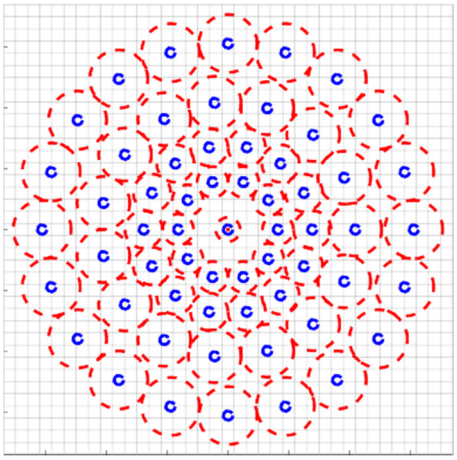
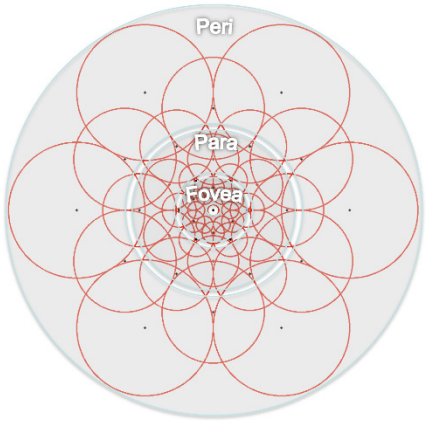


Ilustración 2.24 : Esquema FREAK vs sistema neuronal de la retina, donde los ganglios[26] representan la no linealidad del sistema. [27]

El descriptor consiste en un vector binario al igual que BRISK [11] , ORB [14] y BRIEF [10] .

La metodología BRIEF consiste básicamente en *samplear* el área que rodea el punto de interés, para luego hacer **512 comparaciones** entre dichos puntos y así construir un **vector binario** de 512 componentes. Luego en ORB y BRISK, se usa la misma metodología para construir el vector binario, pero lo que cambia es el **patrón de muestreo** del área. Los patrones de muestreo están representado en la Ilustración 2.25.





[26] Los ganglios son puntos de relevo o de conexiones intermedias entre diferentes estructuras neurológicas del cuerpo, tales como el sistema nervioso central y los fotoreceptores en este caso.

[27] Imagen tomada de [13].

Ilustración 2.25: A la izquierda el patrón de muestreo BRISK[28] a la derecha el patrón de muestreo retinal[29].

Estos patrones de muestreo se encuentran asociados a una escala *s* particular y por lo tanto permiten su lograr invariancia a escala usando distintos patrones, similar a SURF.

El vector binario F está compuesto de una secuencia de respuestas de un bit, este vector corresponde al descriptor BRISK, o FREAK.

$$F = \sum_{0 \le a \le N} 2^a T(P_a)$$

$$Donde\ a\ representa\ el\ a - esimo\ elemento\ binario\ 2^a$$

$$P_a : Un\ par\ de\ campos\ receptivos^{30}$$

$$N: Largo\ deseado\ del\ descriptor\ binario.$$

Y la respuesta al **campo receptivo** es:

$$T(P_a) = \begin{cases} 1\ si\ I(P_a{}^1) - I(P_a{}^2) > 0 \\ 0\ si\ no \end{cases}$$

$$I(P_a{}^1): Intensidad\ del\ primer\ elemento\ del\ par,$$
$$suavizado\ por\ una\ gaussiana\ centrada\ en\ el\ punto\ de\ interés.$$

Con un par de docenas de campos receptivos, miles de parejas pueden ser generadas, sin embargo estas pueden terminar siendo altamente correlacionadas y poco discriminativas.

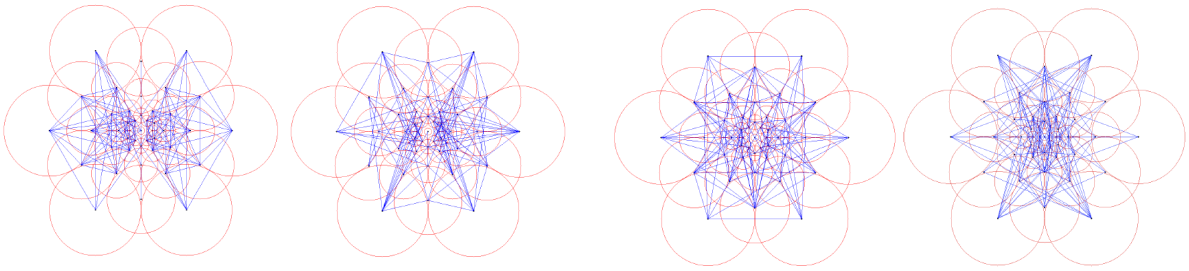
Para poder generar un conjunto de pares de campos receptivos altamente discriminativo en FREAK [13] se realiza un entrenamiento sobre una base de 50.000 puntos de interés, para poder así elegir un conjunto de parejas de campos receptivos que mantengan una varianza alta.

De los 43 campos receptivos de la Ilustración 2.25 existe un total de 903 pares posibles, de los cuales se eligen 512 pares de campos receptivos que forman parte del descriptor FREAK, que se pueden observar en la Ilustración 2.26.

[28] Imagen obtenida desde [10].

[29] Imagen obtenida desde [13].

[30] Un campo receptivo es un punto muestreado alrededor del punto de interés, para generar el descriptor, análogo los campos receptivos de la retina.



a) b) c) d)

Ilustración 2.26 : Pares de campos receptivos de FREAK[31]

Estos **campos receptivos** son interpretados por los autores del método como las distintas respuestas de la retina. Dentro de la Ilustración 2.26 en los patrones de las figuras a) y b) predominan la respuesta perifoveal, más gruesa y en c) y d) la respuesta foveal, más fina. Es decir c) y d) integran información más detallada que a) y b) por lo que el sistema se puede aplicar en cascada.

Asignación de orientación

La asignación de orientación se hace estimando el gradiente local sobre un grupo selecto de pares de campos receptivos G (Ilustración 2.27):

$$O = \sum_{P_o \in G} \left(I\left(P_o{}^1\right) - I\left(P_o{}^2\right)\right) \frac{P_o{}^1 - P_o{}^2}{\left\|P_o{}^1 - P_o{}^2\right\|}$$

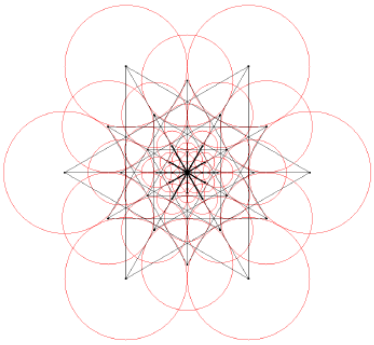


Ilustración 2.27 : Conjunto G de pares de campos receptivos. [32]

De esa manera queda calculado un descriptor FREAK para una imagen I, cabe destacar que FREAK no describe una estrategia para detectar puntos de interés, por lo que debe ser acoplado a un detector de puntos interés por ejemplo SURF.

[31] Imagen obtenida desde [13].

[32] Imagen obtenida desde [13].

$$d_i{}^I \in D, D^I: Puntos\ de\ interes\ de\ la\ imagen\ I.$$

$$d_i{}^I = \begin{pmatrix} d_i{}^I(1) \\ \vdots \\ d_i{}^I(512) \end{pmatrix} = \sum_{0 \le k \le N} 2^i d_i{}^I(k)\,, vector\ de\ 512\ bits.$$

Calce de descriptores

El calce de descriptores binarios es análogo al de descriptores no binarios, pero en vez de usar la distancia euclidiana para calcular los vecinos, es posible usar la **distancia de hamming** [33]que es mucho menos costosa computacionalmente.

Como los pares de campos descriptivos tienen una jerarquía de grueso a fino, el calce se puede hacer en cascada, descartando los descriptores mediante los primeros 16 bytes (Ilustración 2.26 a)). Por eso FREAK es una metodología de cálculo y calce de descriptores **extremadamente eficiente**.

2.2.3 Reconocimiento de color.

Todas las metodologías usadas anteriormente son basadas en la **intensidad** de la imagen, considerando una imagen monocroma que no contiene más que un canal de información.

Es de particular interés para la detección de logos el integrar la información de color de la imagen ya que al pasar una imagen de blanco y negro a RGB, se puede perder información de forma y por lo tanto posibles descriptores que podrían ser determinantes a la hora de detectar el logo.

En la Ilustración 2.28 se muestra un ejemplo de cómo bordes y esquinas bien definidas en el espacio RGB (a), son suavizadas al pasar a escala de grises(b) usando la función rgb2gray[34] de *matlab*, mientras que el canal R (c), conserva los bordes de la imagen RGB.

(a) (b) (c)

Ilustración 2.28 Perdida de información de bordes al pasar a escala de grises.

[33] La distancia de hamming corresponde al número de bits diferentes entre dos vectores binarios.

[34] http://www.mathworks.com/help/images/ref/rgb2gray.html

Este problema puede ser crítico, ya que es justamente en logos donde se pueden encontrar estos cambios abruptos de color que no suelen ser comunes en otro tipo de ambientes, por ejemplo una fotografía.

El tema de los descriptores de color ha sido abordado por Van der Sandee en [15], donde se presenta una serie de enfoques para generar descriptores de color.

2.2.3.1. Representaciones de color

La **representación RGB** suele no ser la mejor para representar los histogramas de color u otros descriptores de color, para ello las representaciones **HSV** y **HSL** permiten hacer mejor uso de la información de color, representados como cilindros en la Ilustración 2.29. HSV usa los canales *hue, saturation* y *value*, mientras que HSL usa los canales *hue saturation* y *lightness*.

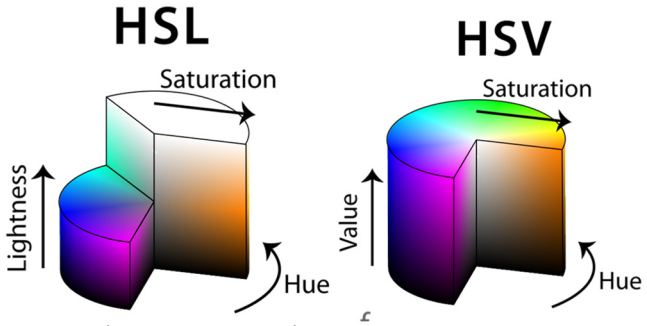


Ilustración 2.29: Espacios de color HSV y HSL[35]

2.2.3.2. Histogramas de color:

En procesamiento de imágenes, el histograma de color es una representación de la distribución de colores dentro de una imagen o parche. En imágenes digitales el histograma representa la **cantidad de pixeles** que tienen un color dentro de una lista fija de rangos de color, que representan un espacio de color.

		ROJO			
		0-63	64-127	128-191	192-255
AZUL	0-63	**43**	78	18	0
	64-127	45	67	33	2
	128-191	127	58	25	8
	192-255	140	47	47	13

Ilustración 2.30: Ejemplo histograma de color en rojo y azul, dividiendo los 256 valores de cada color en 4 *bines*

En la Ilustración 2.30 se muestra un ejemplo de histograma en dos canales, cada canal es muestreado en 4 *bines* que agrupan los 256 valores posibles para cada pixel. Luego se

[35] Obtenido desde [31].

hace un conteo y se genera el histograma. Por ejemplo en Ilustración 2.30 se puede observar que la imagen tiene 43 pixeles con ambos canales rojo y azul entre 0 y 63.

Se puede hacer reconocimiento de objetos simplemente usando calce de histograma, es decir calculando la diferencia entre los histogramas de la imagen *template* y la imagen *sample* y discriminando según su verosimilitud.

Histogramas:

El **histograma RGB** es la combinación de los tres histogramas unidimensionales R, G, B.

El **histograma oponente** es la combinación de los tres histogramas basado en el espacio de color oponente, que queda descrito con los canales O_1, O_2, O_3:

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix}$$

La intensidad queda representada en el canal O_3 y la información de color en los canales O_1 y O_2, la representación oponente suele ser mucho más robusta que RGB, dicha representación además entrega invariancia a iluminación, al iluminar la imagen con luz blanca los canales O_1 y O_2 no se ven afectado ya que al iluminar ambos canales con una fuente ε se cumple que:

$$\frac{(R+\varepsilon)+(G+\varepsilon)-2(B+\varepsilon)}{\sqrt{6}} = \frac{R+G-2B}{\sqrt{6}}$$

El **histograma *Hue*** se basa en el espacio de color **HSV**, donde cada muestra de *Hue*, es ponderada por la saturación porque el canal *hue* es inestable cuando la saturación es cercana a 0. En la Ilustración 2.31 se puede ver como el canal H es inestable en los puntos donde la saturación es muy baja

Esta corrección hace que el histograma sea invariante a distorsiones e iluminación según Van der Weijer en [16].

RGB Canal H (*Hue*) Canal S (*Saturacion*) Canal V(Brillo)

Ilustración 2.31 : Logo *IntellMEDIA* en representación HSV, canal H inestable.

El **histograma RG** consiste en el histograma RGB normalizado, lo cual entrega cierta invariancia a cambios de iluminación y sombras, se llama RG ya que b es redundante, $r + g + b = 1$;

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} = \begin{pmatrix} \frac{R}{R+G+B} \\ \frac{G}{R+G+B} \\ \frac{B}{R+G+B} \end{pmatrix}$$

La **distribución de color transformado** consiste en normalizar el histograma RGB por canal, es decir:

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} \frac{R-\mu_R}{\sigma_r} \\ \frac{G-\mu_G}{\sigma_G} \\ \frac{B-\mu_B}{\sigma_B} \end{pmatrix}$$ a

μ_R: *Media del canal R en la imagen.*

σ_r: *Varianza del canal R en la imagen.*

2.2.3.3. Clasificación por histograma

Una manera simple de detectar objetos por color es mediante la distancia euclidiana entre los histogramas, de la misma manera en que se calza el histograma de gradientes en SIFT.

Un clasificador débil de color se puede implementar de la siguiente manera:

- **Recorrer ventanas** en la imagen *sample*.
- Calcular el **histograma de color** de n *bines*.
- Se **normaliza** el histograma.
- Se calcula la **distancia** entre el histograma del *template y sample*.
- **Clasificar** usando como métrica la distancia calculada.

2.2.3.4. SIFT en color.

El método SIFT descrito por Lowe [8], usa un histograma de orientación de la intensidad como descriptor del parche que rodea el punto de interés, pero no integra

información de color, en [15] se plantean numerosas maneras de aplicar la metodología SIFT usando histogramas de color.

HSV-SIFT, planteado por Bosh en [17] , computa SIFT sobre los 3 canales HSV, en vez de usar solo la intensidad, esto entrega un descriptor de tamaño 3x128, donde la etapa de calces puede ser aplicada sobre cada canal y la validación sobre todos los calces de los 3 canales, el problema de HSV es que ante baja saturación, el valor de *hue* es inestable.

***HueSIFT*,** planteado por Van de Weijer en [16], concatena el histograma *hue* al descriptor SIFT, si el histograma *hue* tiene 128 valores, *HueSift* genera un descriptor de largo 128+128=256. Se plantea como una solución a la inestabilidad de la saturación en HSV-SIFT.

***OpponentSIFT* o SIFT oponente** calcula SIFT normal y concatena al descriptor los histogramas en las bandas $O_1\ y\ O_2$ del espacio de color oponente, de esa manera se obtiene un descriptor invariante a iluminación, ya que tanto sift como las bandas $O_1\ y\ O_2$ son invariantes a iluminación. Corresponde a un descriptor de largo 128+128+128=384 bytes.

W-SIFT, se basa en el espacio w planteado por Geusebroek en [18], similar al espacio oponente, pero donde se normalizan los canales $O_1\ y\ O_2$ por la intensidad $\frac{O_1}{O_3} y\ \frac{O_2}{O_3}$.

rgSIFT, Esta estrategia calcula descriptores SIFT para la intensidad y los canales normalizados r y g.

Transformed color SIFT, en esta estrategia se calcula un descriptor SIFT sobre cada uno de los canales normalizados, es invariante a iluminación, distorsión y sombras debido a la normalización de los canales. Dicha normalización es por canal sobre la región de interés en la imagen, es decir en el punto de interés se calcula el siguiente histograma:

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} \frac{R - \mu_R}{\sigma_r} \\ \frac{G - \mu_G}{\sigma_G} \\ \frac{B - \mu_B}{\sigma_B} \end{pmatrix}$$

(μ_T, σ_B): *Media y varianza del canal T alrededor del punto de interes*

Capítulo 3 Diseño del sistema

3.1 Requerimientos básicos

Luego de analizar el contexto técnico del problema, es necesario definir exactamente qué es lo que debe cumplir el sistema.

El sistema debe ser capaz de:

- Detectar **logos** automáticamente.
- Reconocer anuncios **publicitarios**.
- Asignar una **descripción** a dichas publicidades, que incluya los atributos mencionados en la Tabla 3.1.
- Hacer una buena **estimación de precio** del espacio publicitario, basados en la estrategia de tarificación de cada medio.

En la práctica se debe generar una base de datos donde las entradas tienen los siguientes atributos.

Tabla 3.1 Atributos para lograr describir publicidad en diarios.[36]

ATRIBUTO	DESCRIPCIÓN	EJEMPLO	PROBLEMA
SOPORTE	El Medio en el cual se encuentra el espacio publicitario	LA CUARTA	Trivial
DÍA	El día de la semana en el cual fue anunciado	LUNES	Trivial
FECHA	La fecha en cual fue anunciado.	27-05-2013	Trivial
PAGINA	La página del medio.	13	Trivial
SECCIÓN	La sección del diario en la cual fue anunciado.	ESPECTACULO	Segmentar diario por sección.
AVISO	ID del anuncio, los anuncios suelen tener varias repeticiones	SALCOBRAND OFERTAS	Detectar anuncio publicitario.
PRODUCTOS	Los productos asociados al anuncio.	FARMACIA, PRODUCTOS DE HIGIENE, MEDICAMENTOS	Interpretar contenido.
COLOR	Si el anuncio es o no es a color.	NO	Detectar color
EMPRESA	La empresa que anuncia.	SALCOBRAND	Interpretar conenido
RUBRO	El rubro de la empresa	FARMACIAS	Detectar e interpretar logos.

[36] Megatime, la empresa que ofrece el servicio de verificación de anuncios publicitarios en prensa escrita en chile, ofrece una base de datos con atributos similares.

MARCAS	Marcas publicitadas en el anuncio.	SALCO BRAND,PEPSODENT , TAPSIN	Interpretar contenido
VALOR $	Estimación del valor del anuncio publicitario	1.856.612 CLP	Tarificación

Cabe notar que si bien gran parte del proceso parece ser automatizable, mediante estrategias de visión computacional como SURF o Viola-Jones, hay ciertas tareas **difícilmente pueden ser automatizadas** como:

- Identificación de productos asociados.
- Descripción del anuncio.
- Identificación certera de marcas asociadas.

Por lo cual es necesario integrar inteligencia humana al sistema, es decir, existe una etapa del proceso donde **un operador humano** revisa la publicidad, y completa la entrada.

Para ello se diseña la solución que consiste básicamente en separar el problema en distintos **módulos**, que completan la base de datos, representado en la Ilustración 3.1.

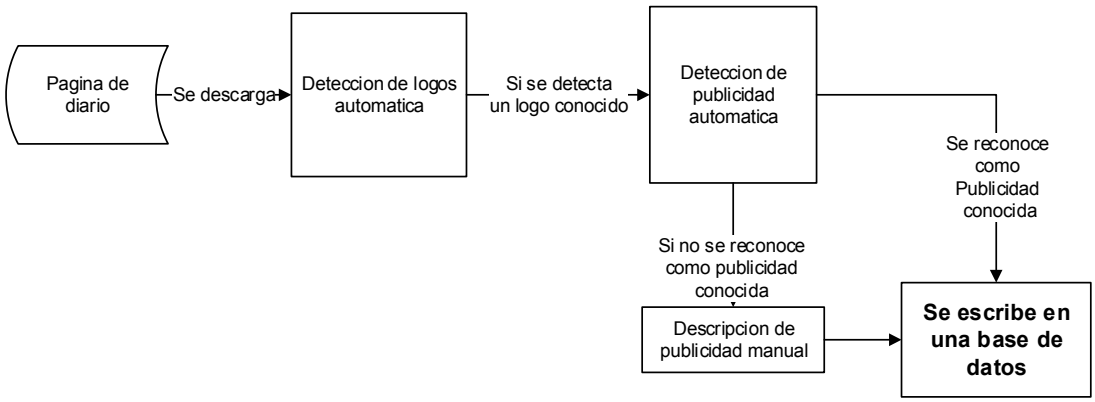


Ilustración 3.1 Descripción básica del sistema

3.2 Adquisición de medios

Generalmente los medios tienen una versión online disponible, un archivo PDF o directamente una imagen, de cada página del diario.

Se ha realizado una recopilación de los medios pertenecientes a la prensa nacional, y la totalidad de los 12 más difundidos cuentan con versiones online descargables.

Tabla 3.2: Principales medios nacionales

Medio	Formato	Adquisición
DIARIO FINANCIERO	PDF	Versión online
EL GRÁFICO	PDF	Versión online
EL MERCURIO	PDF	Versión online
ESTRATEGIA	PDF	Versión online
HOY X HOY	PDF	Versión online
LA CUARTA	PDF	Versión online
LA HORA	PDF	Versión online
LA SEGUNDA	PNG	Versión online
LA TERCERA	PNG	Versión online
LAS ULTIMAS NOTICIAS	PNG	Versión online
PUBLIMETRO	PDF	Versión online
PULSO	PDF	Versión online

Los diarios que cuentan con una versión digital son fáciles de obtener, mientras que para aquellos que no cuentan con una versión online, pueden ser escaneados. Este proceso se muestra en la Ilustración 3.2.

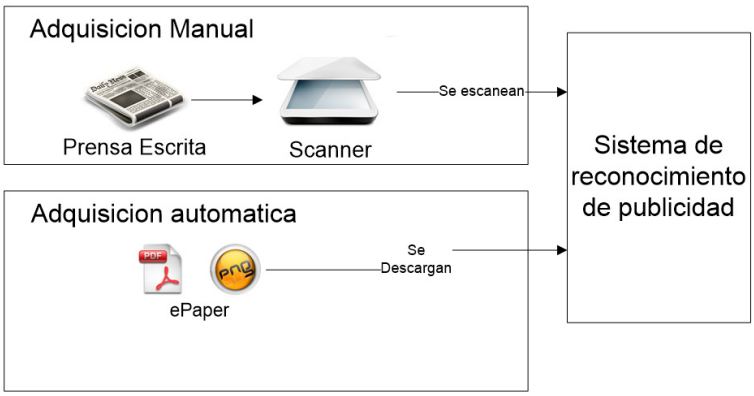


Ilustración 3.2 Adquisición automática y manual.

Cabe destacar que el reconocimiento se hace sobre una imagen representada como mapa de bits, por lo tanto un PDF debe ser convertido a imagen antes de ser procesado. Como existe una pérdida de información al pasar un documento a mapa de bits se debe considerar una resolución donde la distorsión a resolución de 1cm sea aceptable[37].

3.3 Pre procesamiento:

Una página de diario consta básicamente de Cuerpo (texto e imágenes), y publicidad que contiene texto, imágenes las cuales pueden ser logos.

[37] Es un requerimiento que el tamaño mínimo de un logo para ser de interés debe ser 1cm.

Se quiere descartar áreas donde la probabilidad de aparición de un logo sea baja, en particular el cuerpo.

Imágenes del cuerpo del diario:
Se asume a priori que no existen diferencias significativas dentro del diario entre imágenes publicitarias y no publicitarias, es decir, una imagen puede o no contener un logo por lo cual no es simple descartar áreas que contengan imágenes.

3.3.1 Segmentación del texto.

Un diario tiene una tipografía definida, por lo que las zonas de texto del cuerpo del diario pueden ser detectadas y borradas desde el diario. Para ello primero se estudia la tipografía de la base de datos usada en esta memoria.

Se estudia un recorte de LT20000[38] mostrado en la Ilustración 3.3, obtenido de La Tercera viernes 12 de abril 2013, página 26, cuyas dimensiones son 2343 x 3413 pixeles. Si bien consiste en una página, las tipografías son regulares en todo el diario.

Ilustración 3.3 : Recorte LT 12/04/2013

Se pueden identificar tres tipos de tipografía en la imagen, visibles en la Ilustración 3.4:

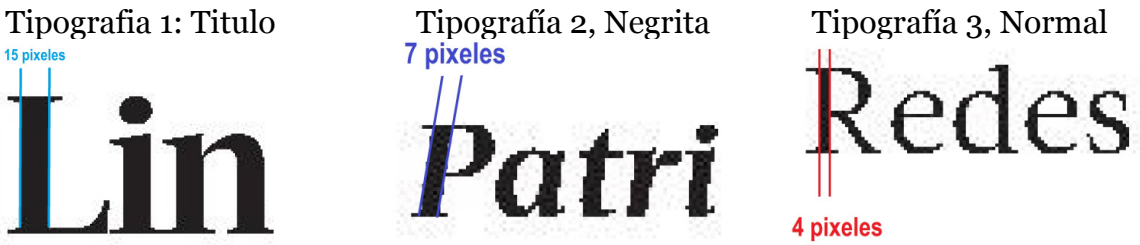


Ilustración 3.4 : Tipografías del diario

[38] Base de datos de diarios de validación.

Para eliminar el texto de la imagen se puede simplemente aplicar una operación morfológica de erosión sobre un elemento estructural de tipo disco, análogo al de la Ilustración 3.5.

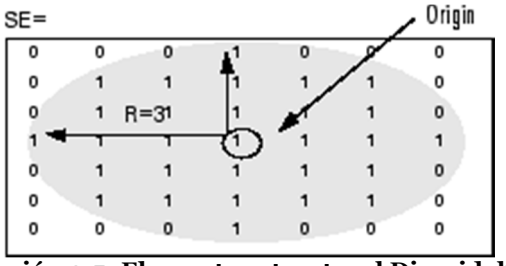


Ilustración 3.5: Elemento estructural Discoidal[39]

Al aplicar el elemento estructural sobre las distintas tipografías se obtiene el resultado de la Tabla 3.3:

Tabla 3.3 : Erosión sobre texto

Tamaño del elemento estructural.	**Tipografía 1**	**Tipografía 2**	**Tipografía 3**
1	Lin	*Patri*	Redes
2	Lin	*Patri*	Redes
3	Lin	*Patri*	
4	Lin	*Patri*	
5	Lin		
6	Lin		
7	Lin		
8	Lin		
9			

[39] Obtenido desde [3].

Se puede observar que para la imagen de 2343 x 3413 pixeles, se puede eliminar completamente el texto mediante el uso de discos de radio 4, 5 y 9, para las tipografías 1,2 y 3 respectivamente.

En la práctica la tipografía 1 es muy gruesa, y puede presentar colisiones con logotipos pequeños, es decir aplicar la operación morfológica borraría partes relevantes de la imagen.

Se puede generalizar el tamaño del filtro en función de la resolución de la imagen, como queda expuesto en la Tabla 3.4.

Tabla 3.4 Tipografías la tercera

Nombre	*Tipografía*	*Radio del elemento estructurante mínimo*	*Tamaño porcentual Con respecto a la resolución*[40]	*Colisión con logos*
Tipografía 1	Lin	9	0,001252	SI
Tipografía 2	*Patri*	4	0,001565	No
Tipografía 3	Redes	5	0,002817	No

Una vez erosionado se aplica la operación morfológica inversa, **dilatación**, usando el mismo elemento estructural, para así reconstruir los segmentos de imagen erosionados.

Esta imagen erosionada y dilatada da origen a una máscara binaria que es usada para segmentar la imagen. En el diagrama de la Ilustración 3.6 se muestra el proceso de borrado de texto.

[40] Tamaño del elemento estructurante en relación al ancho de la imagen de diario.

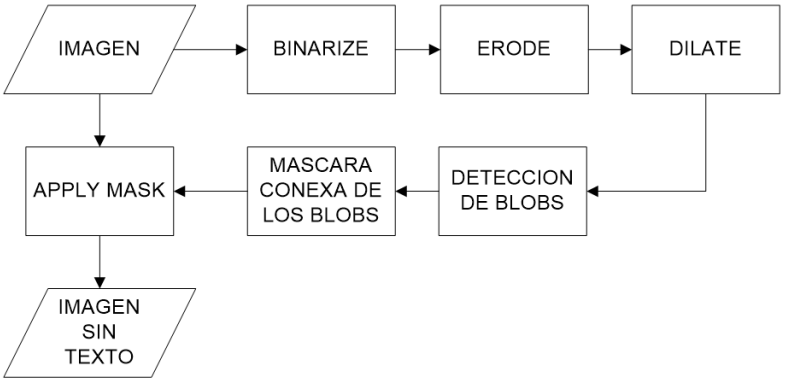


Ilustración 3.6: Preprocesamiento para descartar el cuerpo del texto.

La máscara binaria es generada calculando los blobs de la imagen y luego 'rellenarlos', obteniendo su extensión conexa, de esa manera no hay hoyos en la región y la máscara no recorta texto o detalles al interior de la imagen. Este proceso se puede observar en Ilustración 3.7.

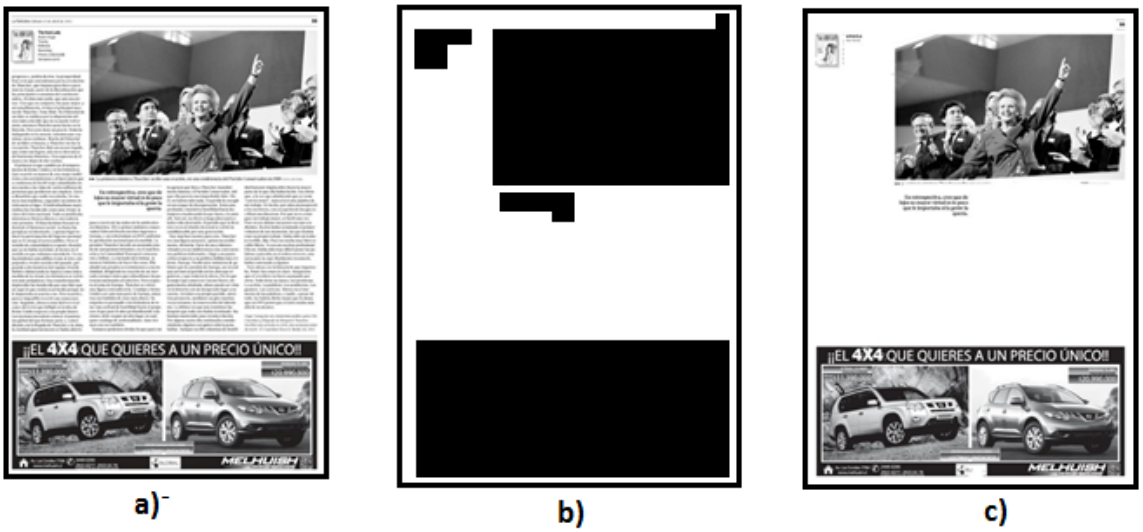


Ilustración 3.7 Preprocesamiento de la imagen, a) Imagen original, b) Mascara de segmentación, c) Imagen segmentada.

Para evaluar el efecto del preprocesamiento en la eficiencia del sistema, se calculan puntos de interés y descriptores SURF en 300 páginas. En la Tabla 3.5 se muestran los tiempos de cálculo y numero de descriptores por página en el experimento.

Tabla 3.5 Tiempo de cálculo imagen preprocesada

	Imágenes Segmentadas	*Imágenes no segmentadas*
Tiempo de cálculo puntos de interés	450 segundos	699 segundos
Tiempo de cálculo descriptores	196 segundos	923 segundos
Numero de descriptores promedio por pagina	15.000	87.000

El costo computacional es reducido notoriamente, un 35% en la etapa de detección y un 80% en la etapa de cálculo de descriptores, esto debido a que justamente las áreas descartadas, con texto, son las que dan origen a gran parte de los **puntos de interés**.

Reducir el número de puntos de interés va de la mano con la precisión del clasificador, un menor número de puntos de interés da origen a menos calces casuales y por lo tanto mejor precisión.

3.3.2 Contextualización:

Contextualizar, definir la ubicación del logo/anuncio en el diario es de suma importancia para lograr estimar con confianza la **tarifa** del anuncio publicitario.

Los parámetros de interés son[41]:

- Día: El precio varía por el día de la semana.
- Paridad: Paginas impares son más caras.
- Portada/Contraportada/Páginas centrales: Dichas páginas son más caras.
- Sección[42]: Existe un factor de precio que varía por la sección del diario.

Identificar el **día**, la **paridad** o si una página es **portada, contraportada o páginas centrales** es trivial. Sin embargo identificar la sección a la cual pertenece no es tan simple y requiere de cierto modelamiento del diario.

Clasificación de sección:

Para lograr segmentar las páginas del diario por sección, se hace uso del encabezado de las páginas.

El diario la tercera cuenta con 11 secciones, que se pueden observar en la Ilustración 3.8:

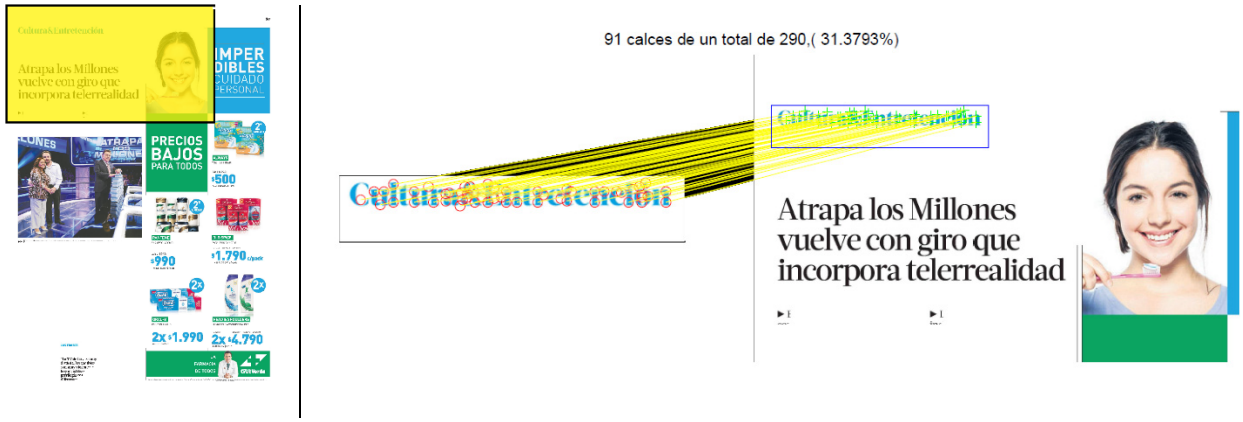
[41] Basados en el tarifario La tercera 2014 [2].

[42] País, mundo, tendencias, deportes…etc…

Ilustración 3.8 Encabezados de La Tercera.

El problema consiste básicamente en **OCR**[43] , pero como ya se tiene una descripción gráfica precisa de los caracteres resulta más conveniente usar detector de encabezados basado en SURF, que actúa sobre un **área precisa de la imagen** y no debe ser necesariamente invariante a rotaciones, pero si a escala.

Se procesa cada una de las páginas del diario, detectando los doce encabezados sobre **un área de interés** de la imagen, un área rectangular ubicada en (x,y)=0,0 y (w,h)=($\frac{3}{4}w_{pagina}$,$\frac{1}{4}h_{pagina}$) , representado por el **área amarilla** en la Ilustración 3.9.



a) b)

Ilustración 3.9 Detector de encabezados

Como no todas las páginas contienen un **encabezado** que indique la sección, aquellas páginas donde no se encuentra un encabezado quedan ubicadas en la última sección para la cual se encontró un encabezado. Es decir si la pagina 25 es clasificada como 'País', y las páginas 26 y 27 no tienen calces en el área de interés, se considera que las páginas 26 y 27 también pertenecen a la sección 'País'.

Se hace una prueba haciendo calce de descriptores SURF entre 110 páginas que contienen los encabezados de la Ilustración 3.8 logrando detección perfecta, si bien solo 10 *samples* por cada encabezado no es una muestra estadísticamente relevante, esta prueba de conceptos muestra que SURF es una excelente herramienta para resolver este

[43] **OCR** es el Reconocimiento Óptico de Caracteres.

problema. Queda planteado como trabajo futuro hacer una prueba más detallada sobre el desempeño.

Tabla 3.6 Detección de encabezados.

$T_{encabezado}$[44]	**20%**
Vp	1
Fp	0
Precisión	1
Recall	1

El **preprocesamiento** presenta un problema logístico si se quiere implementar el sistema, cada diario tiene tipografías y encabezados distintos, además estos cambian cada cierto tiempo. Por ejemplo, la sección 'País' de la Ilustración 3.8 desde el año 2014 pasó a llamarse 'Nacional', por lo que el detector de encabezados dejaría de ser válido.

3.4 Detección de logos.

La primera etapa para poder detectar anuncios publicitarios es detectar logos dentro de la imagen, el problema corresponde a la clasificación de imágenes no binaria donde se quiere reconocer en las imágenes uno de N_L logos sobre un conjunto de páginas P:

Se quiere obtener un clasificador C que detecte la posición correcta de los logos $l_i \in L$ en un conjunto de paginas P.

$$C(P) = \left\{ \begin{bmatrix} (x_1, i_1) \\ \vdots \\ (x_k, i_k) \\ \vdots \\ (x_n, i_n) \end{bmatrix} : si\ el\ logo\ l_{i_k}\ es\ detectado\ en\ la\ posicion\ x_k \right.$$

Se pueden generar N_L clasificadores independientes, que funcionen independientemente para cada logo, aunque es conveniente usar un esquema *bag of words*[45], que permita hacer la detección de manera más eficiente y así reducir el orden computacional del clasificador. Para lograr un buen desempeño es conveniente agregar estrategias en cascada que permitan entregar alta precisión al clasificador sin sacrificar eficiencia.

Aunque cabe destacar que este sistema no tiene requerimientos de respuesta en tiempo real, y por lo tanto el costo computacional no es prioridad.

A continuación se describen los pasos para lograr detectar un conjunto de logos L dentro de un conjunto de páginas P.

[44] Umbral de porcentaje de *inliers* en el total de puntos de interés de la imagen template para considerar un calce valido,

[45] En visión computacional, el modelo *bag-of-words* o bolsa de palabras, es aplicado para clasificación de imágenes, tratando las características de la imagen como una lista de palabras, descriptores.

1. Se pre calculan los **puntos de interés** y **descriptores** para $l_i \in L$
2. Para cada página de diario $p_k \in P$:
    a. Se calculan los puntos de interés y descriptores de p_k .
    b. Para cada $l_i \in L$
        i. Se calzan los descriptores de ambas imágenes.
        ii. Se calcula la mejor homografía H, usando Ransac.
        iii. Se calcula el score de la homografía

$$score = \frac{Calces_{validados}}{Puntos\ de\ interes\ de\ \ l_i}$$

        iv. Si el score del calce es mayor a T:
            1. Se genera un match entre la pagina P y el logo l_i con la posición estimada descrita por la homografía $m = (l_i, p_k, x)$.
            2. Se repite b. y se vuelve a buscar el mismo template descartando los calces ya validados.
3. Pasa a la siguiente página del diario

Cabe destacar que para lograr encontrar más de una instancia dentro de una imagen, se debe recalcular la homografía omitiendo los puntos de interés que generaron la primera homografía dentro de la imagen sample.

En la Ilustración 3.10 se hace una representación gráfica del calce de logos. Este problema es estudiado en profundidad en el Capítulo 4.

Ilustración 3.10 : Ejemplo logo calzado en página preprocesada.

3.5 Detección de publicidad

El objetivo de este trabajo es diseñar una estrategia de **detección de anuncios publicitarios**, por eso, una vez encontrado un calce de logo dentro de una página,$m_k = (l_i, p_j, \mathrm{x_j})$ es necesario evaluar si dicho logo es parte de un anuncio publicitario, para eso además de contar con un conjunto de logos conocidos L , existe para cada $l_i \in L$ un conjunto de anuncios $a_k \in A^i$ que lo contiene.

$$A^i: Anuncios\ que\ contienen\ al\ logo\ l_i$$

Un **anuncio** corresponde a una imagen rectangular que promociona una marca o producto, el anuncio en la página de diario tiene un tamaño definido modularmente, es decir existen 26 tamaños posibles que pueden ser observados en la Ilustración 3.11 b), donde el anuncio de la figura corresponde al módulo MD3x10, marcado en un rectángulo amarillo en la figura b).

a) b)

Ilustración 3.11 Partes básicas de una página de diario.

Una vez encontrado un match $m = (l_i, p_k, x)$, este debe generar una instancia de anuncio, **recuadro amarillo** en Ilustración 3.11 a), para ello se requiere una etapa de detección de publicidad que queda descrita en el siguiente recuadro:

Cuando se identifica un logo l_i en la página p, se analiza la página del diario, comparando los anuncios de A_{l_i}, los anuncios que contienen al logo l_i.

1. Se leen los puntos de interés e descriptores previamente calculados de la página.
2. Para cada anuncio que contiene l, $a_k \in A_l$
    a. Se calculan o leen los descriptores de a_k:
    b. Se calza y genera la homografía H
    c. Si score>$T_{anuncio}$ y H no es proyectiva.
        i. Se obtiene los límites del anuncio mediante la homografía.
        ii. Se corrigen los límites para que calcen con un módulo publicitario.
        iii. Se escribe en la base de datos la aparición del anuncio.
3. Si no existe ningún anuncio con score>$T_{anuncio}$, la página pasa a ser chequeada por un operador.

Cabe destacar que en este punto se requiere invariabilidad a escala y rotación[46], no transformaciones de perspectiva, ni escalamientos desproporcionados, además como un anuncio es una imagen mucho más compleja genera miles de puntos de interés, lo que hace que la clasificación por descriptores locales sea extremadamente precisa.

Para evaluar el desempeño se testea la **detección de publicidad** con 100 páginas obtenidos de LT20000 con 100 publicidades, se realiza el calce usando SURF obteniendo el siguiente comportamiento:

$T_{anuncio}$	**20%**
Vp	1
Fp	0
Precision	1
Recall	1

Si bien la prueba de conceptos es sobre un conjunto de testeo reducido, que no tiene validez estadística. Demuestra que usar calce de descriptores SURF, en una imagen compleja, con miles de descriptores, en un ambiente libre de distorsiones y perspectivas,

debiese tener un bajísimo nivel de falsos positivos. Queda planteado como trabajo futuro hacer una prueba más detallada sobre el desempeño de detección de anuncios completos.

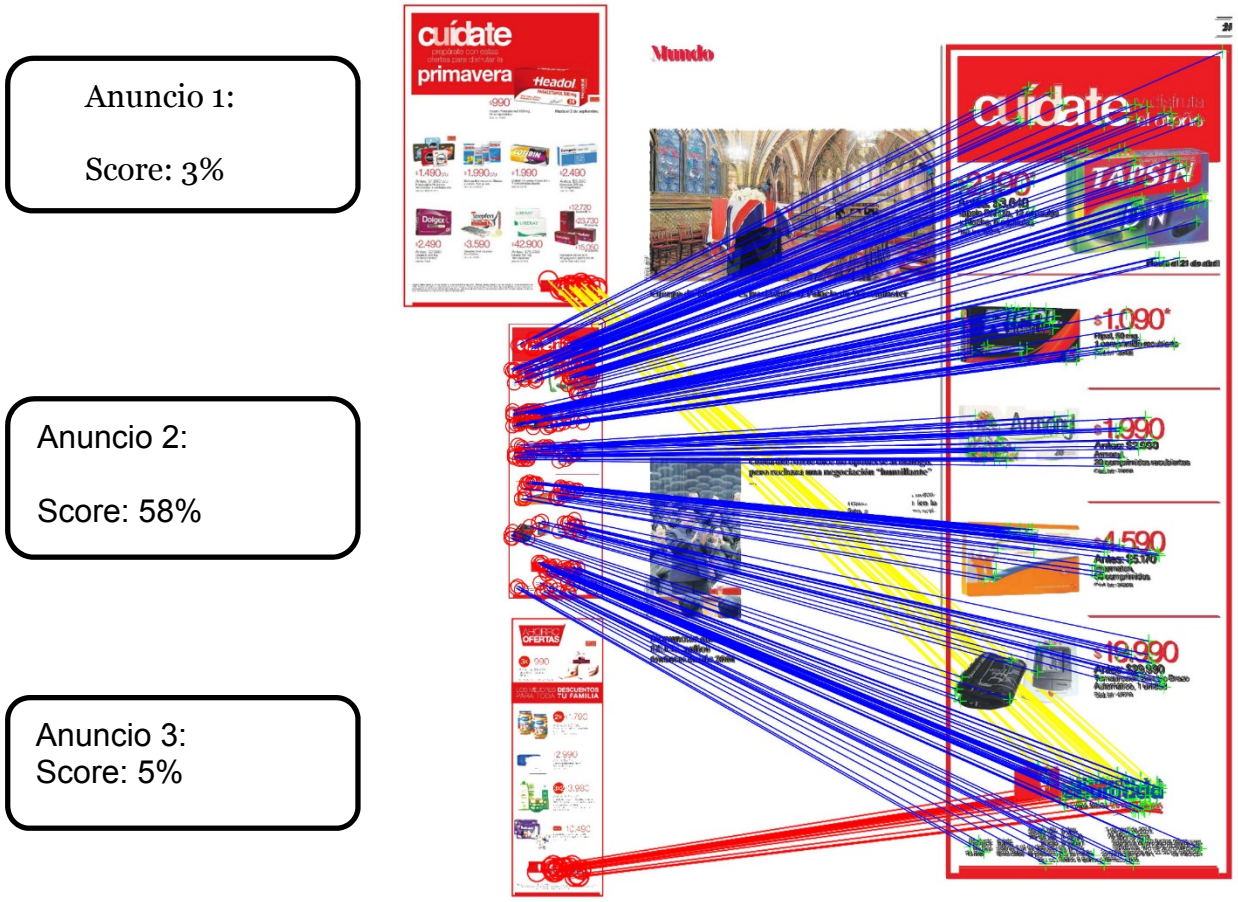


Ilustración 3.12 Detector de anuncios publicitarios

En la Ilustración 3.12 se muestra el proceso de detección de anuncios publicitarios, a la izquierda se encuentran tres anuncios de $A_{ahumada}$, los anuncios que contienen el logo de ahumada, al calzar las imágenes todos generan una homografía valida, ya que contienen el logo 'ahumada', sin embargo es muy fácil distinguir entre el anuncio correcto, ya que tendría un altísimo *score* ($a_{bestscore}$=58%, anuncio 2 en la Ilustración 3.12).

El cálculo de la homografía permite encontrar un rectángulo que define los límites del anuncio $a_{best-score}$ que fue detectado en la página. Las dimensiones de este rectángulo son identificadas como un **módulo publicitario** de los descritos en la Ilustración 3.12.

En el caso particular mostrado en la Ilustración 3.12 :

$$BoundingBox = (x, y, w, h) = (1205,91,1033,3301)$$

Las dimensiones del rectángulo con respecto a las de la imagen (2343x3414).

$$w' = \frac{1033}{2343} \approx \frac{3}{6}$$
$$h' = \frac{3301}{3414} \approx \frac{10}{10} \rightarrow \boldsymbol{Modulo = MD10x3}.$$

Dentro del rectángulo que define los límites del anuncio encontrado pueden existir varias instancias de logos detectados, estos no dan origen a un nuevo anuncio ya que el logo pertenece al anuncio ya encontrado.

En el rubro publicitario se habla de ***share of investment*** y ***share of voice***[47], que corresponde a la marca que publicita, y las marcas presentes en el anuncio, respectivamente. Reconocer que marca tiene el *share of investment* no es trivial, por ejemplo en la Ilustración 3.12 tanto 'Ahumada' como 'Tapsin' tienen presencia en el anuncio, es posible inferir es pagado por 'Ahumada', pero es muy complejo automatizar esta etapa del proceso.

La ambigüedad de quien es el *share of investment* y *share of voice* de un anuncio es una de las principales razones por las que el sistema no es 100% automático y requiere la intervención de un operador.

[47] *Share of investment* corresponde a quien paga el anuncio, mientras que *Share of Voice* corresponde a aquellas marcas que tienen presencia a través del anuncio de manera colateral.

3.6 Tarificación Automática

Para hacer la tarificación del anuncio se evalúa la expresión de 2.1.3:

$$(Precio\ modulo\ x\ Factor\ ubicación\ x\ Factor\ día) + \ Factor\ color\ = \ Valor\ anuncio$$

Siguiendo con el ejemplo de la Ilustración 3.12 y considerando la tarificación del diario la tercera expuesta en la sección 2.1.3, el precio del módulo $MD10x3$ es $1.920.915 CLP, precio base del anuncio.

El factor ubicación y factor día se obtiene evaluando las condiciones que definen el precio[48]:

Tabla 3.7 : Tarificación aplicada al ejemplo

			Aplica
Factor ubicación	Crónica Par	2	No
	Crónica Impar (Mundo)	2,5	Si
	Páginas Centrales	3	No
	Contraportada	3,5	No
	Deportes Par	1,7	No
	Deportes Impar	1,8	No
	Espectáculos Par	1,7	No
	Espectáculos Impar	1,8	No
	Cine	1	No
Factor día	Lunes	1	No
	Martes	1	No
	Miércoles	1	Si
	Jueves	1	No
	Viernes	1,15	No
	Sábado	1,7	No
	Domingo	1,6	No
Factor Color	Lunes-Jueves	500000	Si
	Viernes	575000	No
	Sábado	850000	No
	Domingo	800000	No

Por lo tanto el anuncio tiene una tarificación estimada de

$$1.920.915 * 2.5 * 1 + 500.000 = 1.920.915 * 2.5 * 1 = \mathbf{5.302.288\ CLP}$$

[48] La imagen corresponde a la tercera, página 21 del día miércoles 17 de Abril del 2013.

3.7 Descripción de publicidad

En los puntos anteriores se asumió que existe un conjunto A_l conocido de anuncios publicitarios que contienen al logo l. Sin embargo esta base de datos debe ser generada por un operador, ya que cada día aparecen nuevos anuncios publicitarios que deben ser integrados al sistema, esto ocurre precisamente cuando se detecta el logo y no existe un anuncio en A_l que calce correctamente la instancia.

En esta etapa el operador debe:

- Recortar la imagen del diario, definiendo los límites del nuevo anuncio.
- Identificar productos o campañas asociados al anuncio

A pesar de que todos los nuevos anuncios deben ser descritos y recortados por un operador en algún momento, la implementación del sistema es una mejora sustancial en cuanto a la eficiencia comparado con el caso base, donde el operador lee todo el diario y recorta cada uno de los anuncios.

En la siguiente tabla se observa el total de anuncios publicitados en un mes, y cuántos de ellos son diferentes:

Total instancias	**26726**
Total únicos	1958
Cociente	**13.64964249**

Tabla 3.8 Repetitividad de anuncios en prensa escrita.[49]

Eso significa que cada anuncio aparece un promedio de **trece veces** en un mes, por eso existe una gran diferencia entre hacer detección automática de publicidad y detección manual, ya que de esas 13 veces, 1 vez va a ser descrito por el operador, y 12 veces seria detectado automáticamente.

[49] Datos obtenidos desde base de datos de empresa de verificación de publicidad en chile.

Capítulo 4 Desempeño de estrategias

En este capítulo se evalúa y analiza el desempeño de distintas estrategias para detectar logos de una base de logos *L* en un conjunto de páginas *P*.

4.1 Descripción benchmarking

Para evaluar el funcionamiento de distintas estrategias de reconocimiento de objetos se realiza un ***benchmark***, que consiste en distintas pruebas para medir el desempeño de las metodologías usadas, para así comparar configuraciones de parámetros según su capacidad clasificadora.

4.1.1 Bases de datos

Para realizar la evaluación se construyó la base de prueba **LT20000**, que consiste en una serie de 20.000 páginas de diarios recopiladas de la tercera [19] y **logos de prueba** obtenidos desde el portal Logos Chile Vector [1], dedicado a la distribución de logos gráficos con fines académicos.

4.1.1.1. Logos de prueba.

Se hace una selección de 27 logos de prueba, dichos logos se encuentran en el Anexo A:

Tabla 4.1 : Los 27 logos elegidos para las prueba.

'AHUMADA'	'MITSUBISHI'	'FALABELLAL'	'RIPLEY'	'NISSAN'	'LATERCERA'
'CRUZVERDE'	'SKY'	'FIAT'	'CDF'	'CLARO'	'FACEBOOK'
'DERCO'	'AGRUPEMONOS'	'SALCOBRAND'	'PARIS'	'SAMSUNG'	
'EASY'	'ECLASS'	'UNIMARC'	'JUMBO'	'TWITTER Logo'	
'LANPASS'	'ENTEL'	'MOVISTAR'	'CLUBLATERCERA'	'TWITTER Tipográfico'	

La elección se hace tratando de incluir figuras simples, como Facebook, así como figuras más complejas, como el logo de ahumada, los logos se encuentran en el anexo 7.1.

4.1.1.2. Páginas de prueba

Las páginas son obtenidas directamente desde papel digital en formato **PDF**, por lo que deben ser **convertidas en imágenes**, donde inevitablemente existe perdida de información. Por ello es necesario definir un nivel de *ppi*[50] donde los logos a escala de **1cm** no sean fuertemente distorsionados, como se puede ver en la Ilustración 4.1.

[50] *ppi*, *pixels per inch* es una medida de resolución de impresión.

a) Original *pdf* b) 300 *ppi png* c) 150 *ppi png*

Ilustración 4.1 : Logo de 1 cm x 1 cm

Se considera una impresión a 300 *ppi*, lo cual genera una imagen de 2343x3413 pixeles.

La base de datos consiste en **20.542 páginas** de la tercera entre el año 2011 y 2013, las cuales fueron marcadas manualmente buscando apariciones de los **27 logos**, se requiere que cada logo tenga al menos 100 apariciones dentro de la base de datos.

Total paginas revisadas	**20543**
Total instancias	**3261**
Páginas con instancias[51]	**2031**

La base de datos fue marcada manualmente, obteniendo de esa manera un ***groundtruth***, donde se especifican las apariciones de cada uno de los **logos** dentro de la base de datos de **páginas**.

Para realizar el ***groundtruth*** se implementó una herramienta gráfica en *matlab*, programada para generar la base de prueba particular de este proyecto, como se puede observar en la Ilustración 4.2.

Ilustración 4.2 : Herramienta GroundTruthCreator, diseñada para marcar páginas de diario

[51] Aquellas páginas donde está marcada al menos una instancia de un logo en el *groundtruth*.

4.1.2 Experimentos

Se probaron las siguientes estrategias sobre la base de datos

a) Calce directo de templates.
    a. Re escalamiento de la imagen a 1171x1706[52].
    b. Búsqueda de la ventana que calza con el template.
b) SIFT puro[53].
    a. Cálculo de puntos de interés SIFT
    b. Cálculo de descriptores SIFT
    c. Calce de descriptores.
    d. Cálculo de homografía usando RANSAC, se aceptan homografías proyectivas.
c) SURF puro[54].
    a. Cálculo de puntos de interés SURF
    b. Cálculo de descriptores SURF
    c. Calce de descriptores SURF.
    d. Cálculo de homografía usando MLSAC, solo se aceptan homografías afines.
d) FREAK puro.
    a. Cálculo de puntos de interés SURF
    b. Cálculo de descriptores FREAK
    c. Calce de descriptores binario.
    d. Cálculo de homografía usando MLSAC, solo se aceptan homografías afines.
e) *Color transformed SIFT*[55].
    a. Cálculo de puntos de interés HARRIS-LAPLACE
    b. Cálculo de descriptores SIFT-COLOR
    c. Calce de por canal.
    d. Cálculo de homografía usando MLSAC, solo se aceptan homografías afines.
f) Opponent SIFT
    a. Cálculo de puntos de interés HARRIS-LAPLACE
    b. Cálculo de descriptores SIFT-COLOR
    c. Calce de descriptores por canal.
    d. Cálculo de homografía usando MLSAC, solo se aceptan homografías afines.
g) Viola jones[56]
    a. Generación de base de entrenamiento
    b. Entrenamiento del clasificador.
    c. Testeo sobre la base de validación.
    d. Testeo con discriminación por histograma

---

[52] Se escala la imagen ya que hacer calce de imágenes, multi-escala en resolución 2000+ pixeles no resulta viable.

[53] Usando la implementación de la librería *VLFeat*.

[54] Usando el *toolbox computer visión* de *Matlab*.

[55] Usando *ColorDescriptors*, por Van der Sande [15].

[56] Usando *OpenCv*

4.2 Resultados

4.2.1 Caso base

Un ***clasificador ingenuo***[57] seria uno que encuentra el template en toda posicion.

La **precisión** de dicho clasificador seria la proporción entre las áreas de *ROI's*[58] y el total de la base de datos, mientras que el *recall* seria 1.

$$Precision(t) = \frac{AREA_{ROI}}{AREA_{Total}}$$

Considerando que en la base de datos:

$$AREA_{ROI} = 10.227.128\ pixeles^{59}$$
$$AREA_{Total} = 16.055.987.506\ pixeles$$

El mejor desempeño base de un clasificador ingenuo seria:

Precision	*Recall*	*F-Score*
0.00063	1	0.00063

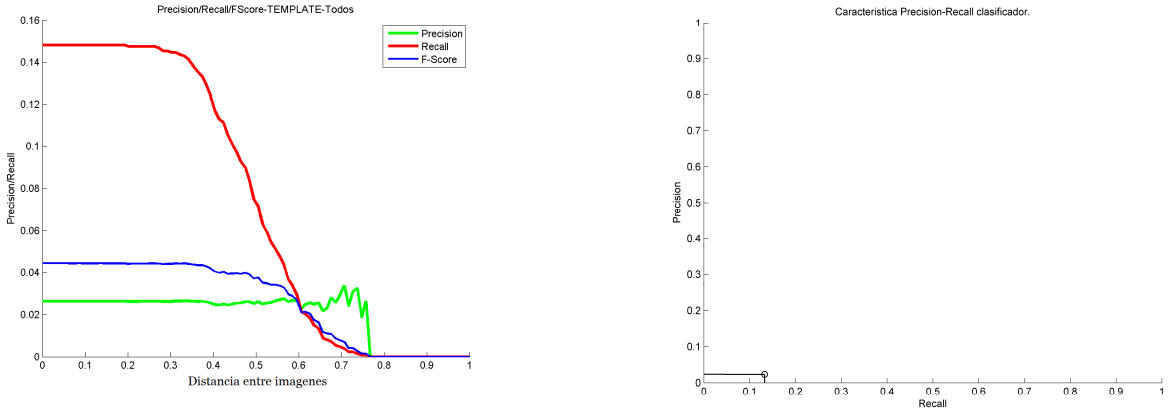
El clasificador ingenuo puede ser considerado como un caso base, es decir si una estrategia de clasificación logra una precisión de 0.00063, realmente no tiene capacidad clasificadora, responde prácticamente al azar.

[57] Un clasificador ingenuo seria aquel que clasifica aleatoriamente.

[58] ROI, región de interés.

[59] Promedio de pixeles, por clase, dentro de todo el GroundTruth.

4.2.2 *Experimento 1: Template matching*:



a) b)

Ilustración 4.3 Desempeño template matching, nótese que el gráfico F-Score no está normalizado entre 0 y 1.

Al calzar directamente las imágenes *samples* y templates, usando la estrategia expuesta en la sección 2.2.2.3 *template based matching*, se obtiene un detector con mínima capacidad clasificadora, como se puede observar en la Ilustración 4.3 .

El algoritmo recorre todos los puntos de la imagen sample, tratando de hacer calzar el template, sobre una única escala. Por eso el algoritmo no es invariante a escala.

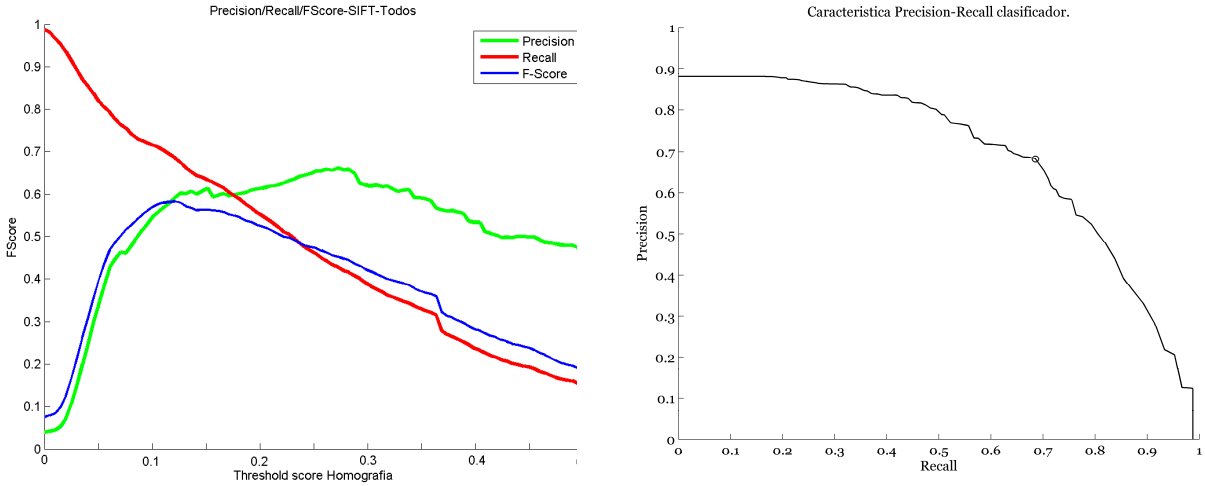
En la Ilustración 4.3 a) se grafica la precisión y *recall* para distintos valores de T, el umbral de clasificación para la diferencia entre la imagen sample y template. Por muy bajo que sea T, el clasificador no es capaz de recuperar todas las instancias en la posición correcta, y solo logra recuperar un 15% de las instancias en el mejor caso.

Si bien el resultado no es decidor, este clasificador es mucho más preciso que el caso base, rondando el 2% de precisión y llegando a un 4% cuando el clasificador es más exigente, a diferencia del caso *naive* que logra una precisión de 0.00063.

Tabla 4.2: Desempeño con el umbral que maximiza f-score para cada logo.

Template	*Precisión*	*Recall*	*F-Score*	*Template*	*Precisión*	*Recall*	*F-Score*
Ahumada	0.05	0.03	0.04	Movistar	0.07	0.05	0.06
Cruz Verde	0.02	0.03	0.03	Ripley	0.03	0.02	0.02
Derco	0.06	0.04	0.05	CDF	0.05	0.03	0.04
Easy	0.03	0.02	0.03	Paris	0.03	0.03	0.03
Lanpass	0.04	0.04	0.04	Jumbo	0.03	0.02	0.02
Mitsubishi	0.14	0.09	0.11	Club LT	0.02	0.01	0.02
Sky	0.02	0.01	0.02	Nissan	0.03	0.02	0.02
Agrupemonos	0.04	0.02	0.03	Claro	0.10	0.08	0.09
Eclass	0.09	0.05	0.07	Samsung	0.10	0.06	0.08
Entel	0.05	0.03	0.04	Twitter Logo	0.01	0.00	0.00
Falabella	0.09	0.09	0.09	Twitter Tipo	0.12	0.09	0.10
Fiat	0.03	0.03	0.03	La tercera	0.09	0.06	0.08
Salcobrand	0.07	0.07	0.07	Facebook	0.00	0.00	0.00
Unimarc	0.04	0.03	0.03	**Promedio**	0.05	0.04	0.05

4.2.3 Experimento 2: SIFT+RANSAC.



a) *b)*

Ilustración 4.4 Desempeño SIFT.

Al hacer calce de descriptores SIFT en la base de datos se obtiene un resultado mucho más acertado**.**

En la Ilustración 4.4 a) se ve que el clasificador, a diferencia del caso anterior, ante un bajo $T, threshold\ de\ homografia,$ presenta un *recall* cercano a 1. Es decir es capaz de identificar la pose correcta de casi todas las instancias, a medida que el T aumenta el *recall* cae, lo que significa que se encuentran los calces correctos, pero en calces con bajos score.

También se puede observar que la precisión es limitada cerca del 0.6, principalmente por el bajo desempeño ante ciertos logos simples (en rojo en la Tabla 4.3), y al hecho de que se aceptan homografías no-afines.

Cabe destacar que en la Ilustración 4.4 a) los puntos de la característica F-Score no corresponden a los puntos del gráfico *Precisión-Recall.* En el gráfico a) se clasifica sobre un T fijo, mientras que en el gráfico b) se promedian los desempeños de cada clasificador, sin fijar un T sino que considerando aquel T que logra el *recall* muestreado en el gráfico, como se puede ver en la Ilustración 4.5.

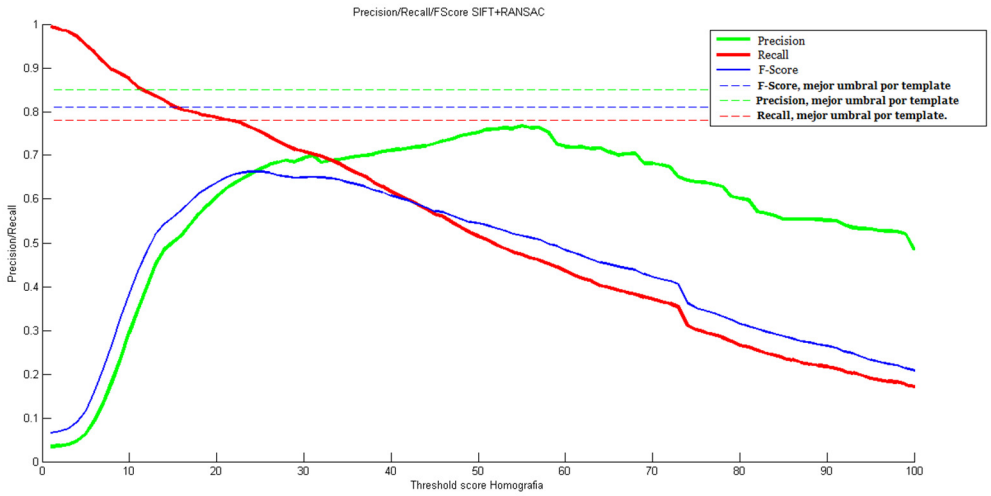
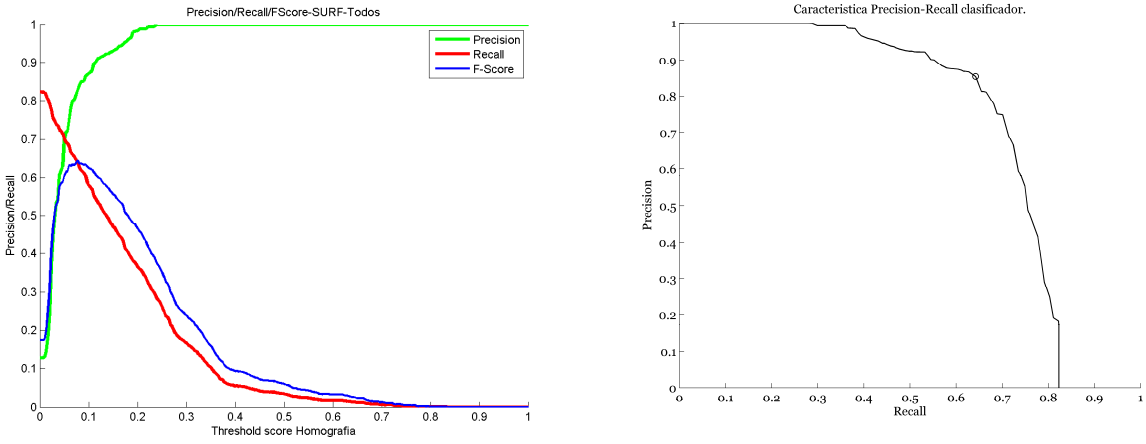


Ilustración 4.5 F-Score umbral fijo vs mejor umbral para cada template.

Tabla 4.3 Desempeño SIFT, con el *threshold* que maximiza F-score para cada logo.

Template	*Precisión*	*Recall*	*F-Score*	*Template*	*Precisión*	*Recall*	*F-Score*
Ahumada	0.98	0.98	0.98	Movistar	0.48	0.66	0.55
Cruz Verde	0.96	0.97	0.96	Ripley	0.94	0.95	0.94
Derco	0.99	0.99	0.99	CDF	0.69	0.79	0.74
Easy	0.74	0.96	0.84	Paris	0.87	0.98	0.92
Lanpass	0.92	0.90	0.91	Jumbo	0.84	0.99	0.90
Mitsubishi	0.96	0.97	0.97	Club LT	0.65	0.84	0.73
Sky	0.99	0.99	0.99	Nissan	0.92	0.98	0.95
Agrupemonos	0.99	0.99	0.99	Claro	0.51	0.36	0.42
Eclass	0.99	0.99	0.99	Samsung	0.87	0.90	0.88
Entel	0.74	0.91	0.82	Twitter Logo	0.40	0.71	0.52
Falabella	0.37	0.37	0.37	Twitter Tipo	0.28	0.21	0.24
Fiat	0.83	0.85	0.84	La tercera	0.39	0.46	0.42
Salcobrand	0.88	0.93	0.91	Facebook	0.29	0.39	0.33
Unimarc	0.33	0.44	0.37	**Promedio**	0.73	0.79	0.76

4.2.4 Experimento 3: SURF +MLSAC



a) b)

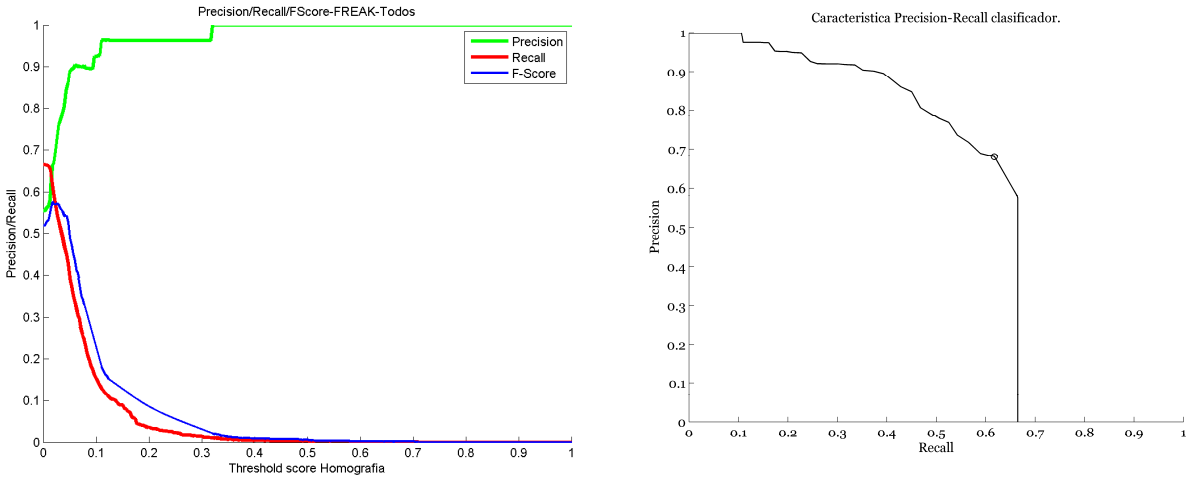
Ilustración 4.6: Desempeño SURF

En este experimento (Ilustración 4.6) se usa el cálculo de **homografía MLSAC**, configurada para rechazar homografías no afines, las cual conllevan a errores como se discute en la siguiente sección. Por ese motivo se logra una precisión de 100% en condiciones restrictivas, a diferencia del experimento anterior. Sin embargo no se logra el ~100% de *recall*, existen homografías que no son encontradas bajo ninguna circunstancia, debido también a restringir el clasificador a homografías afines.

Tabla 4.4 Desempeño SURF con el threshold que maximiza F-Score para cada logo.

Template	*Precisión*	*Recall*	*F-Score*	*Template*	*Precisión*	*Recall*	*F-Score*
Ahumada	0.97	0.98	0.97	Movistar	0.51	1.00	0.68
Cruz Verde	0.98	1.00	0.99	Ripley	0.93	1.00	0.96
Derco	0.99	1.00	0.99	CDF	0.28	0.20	0.23
Easy	0.48	0.41	0.44	Paris	0.69	0.92	0.79
Lanpass	0.98	0.98	0.98	Jumbo	0.59	1.00	0.74
Mitsubishi	0.95	1.00	0.98	Club LT	0.88	0.92	0.90
Sky	0.99	1.00	1.00	Nissan	0.92	0.99	0.96
Agrupemonos	0.99	1.00	0.99	Claro	0.94	0.97	0.95
Eclass	0.99	1.00	0.99	Samsung	0.82	0.96	0.88
Entel	0.72	0.91	0.80	Twitter Logo	0.10	0.46	0.16
Falabella	0.86	0.82	0.84	Twitter Tipo	0.18	0.73	0.29
Fiat	0.67	1.00	0.80	La tercera	0.43	1.00	0.60
Salcobrand	0.89	0.99	0.94	Facebook	0.18	0.23	0.20
Unimarc	0.64	0.58	0.61	**Promedio**	0.72	0.85	0.78

4.2.5 Experimento 4: FREAK+MLSAC



a) b)

Ilustración 4.7 Desempeño Freak

Al evaluar **FREAK**, se obtiene un desempeño menos preciso y menos exhaustivo que con SIFT o SURF, aun cuando se usa MLSAC.

En la Ilustración 4.7 se observa que la cantidad de instancias que no son recuperadas bajo ninguna configuración es aún mayor, cercana a 30%.

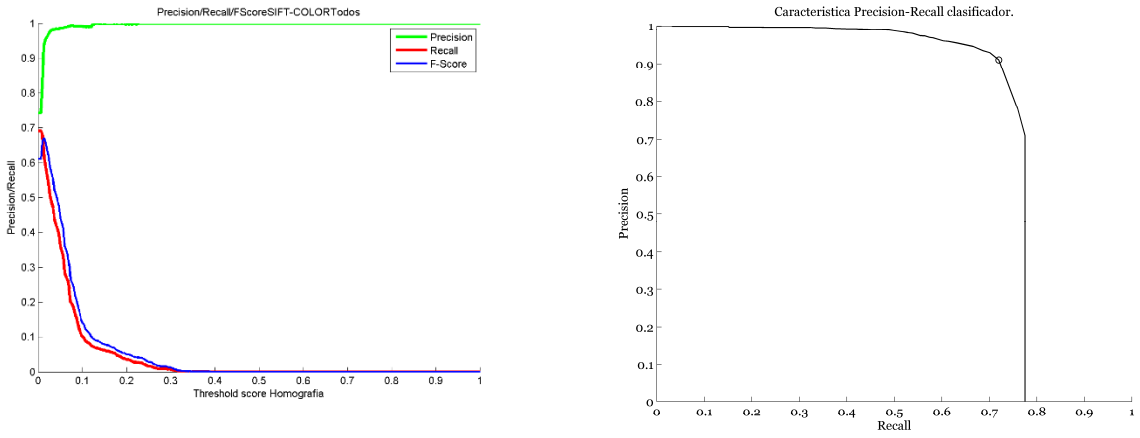
Freak es una estrategia cuyo diseño se basa en lograr eficiencia más que eficacia, sin embargo. Alahi plantea en [13] que el algoritmo tiene repetitividad[60] y eficacia cercana a SURF y SIFT, dentro de su propio *benchmark*. Sin embargo en [20] se plantea que no es correcto y el desempeño es ligeramente inferior, como parece ser el caso.

Tabla 4.5 Desempeño Freak

Template	*Precisión*	*Recall*	*F-Score*	*Template*	*Precisión*	*Recall*	*F-Score*
Ahumada	0.82	0.99	0.89	Movistar	0.39	0.90	0.55
Cruz Verde	0.97	1.00	0.98	Ripley	0.69	0.70	0.70
Derco	0.92	0.99	0.95	CDF	0.40	0.51	0.45
Easy	0.39	0.66	0.49	Paris	0.73	0.87	0.79
Lanpass	0.76	0.79	0.77	Jumbo	0.53	0.63	0.58
Mitsubishi	0.73	0.94	0.82	Club LT	0.69	0.66	0.68
Sky	0.93	1.00	0.96	Nissan	0.87	0.94	0.90
Agrupemonos	0.98	0.98	0.98	Claro	0.82	0.95	0.88
Eclass	0.99	1.00	0.99	Samsung	0.63	0.97	0.77
Entel	0.59	0.58	0.58	Twitter Logo	0.19	0.56	0.28
Falabella	0.70	0.90	0.79	Twitter Tipo	0.03	0.55	0.05
Fiat	0.66	0.85	0.74	La tercera	0.39	0.27	0.32
Salcobrand	0.65	0.60	0.63	Facebook	0.03	0.71	0.06
Unimarc	0.71	0.88	0.79	**Promedio**	0.64	0.79	0.71

[60] Repetitividad es equivalente la robustez en descriptores, descriptores repetitivos generan clasificadores robustos.

4.2.6 Experimento 5: SIFT de color transformado + MLSAC



a) b)

Ilustración 4.8 desempeño SIFT color transformado

Usando **SIFT de color transformado**, en la implementación de Van de Sande, se logra un mejor f-score que SIFT, pero a un menor *recall*.

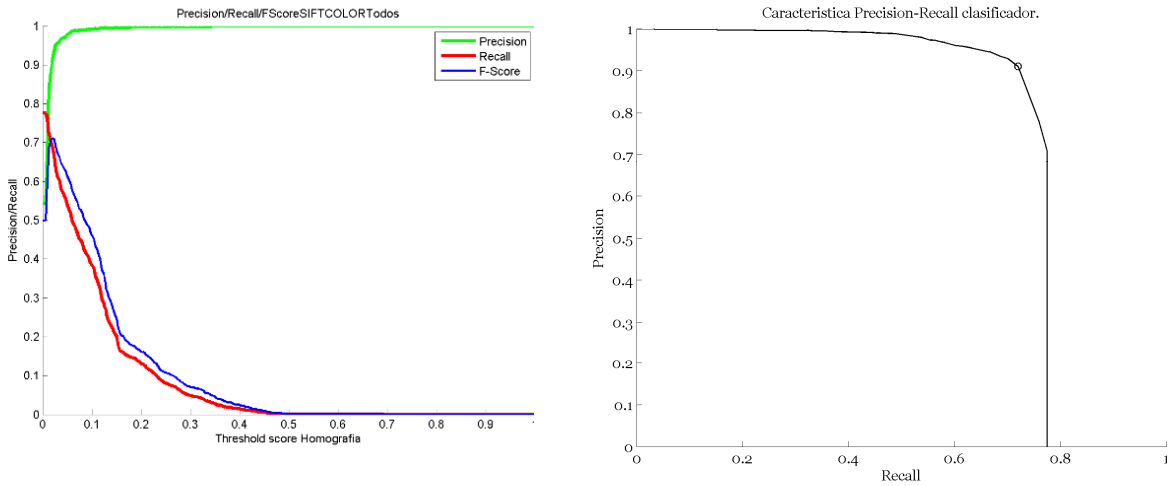
El usar un descriptor de 512 bytes hace que la etapa de calces sea mucho más exigente como se puede observar en Ilustración 4.8 a), donde se ve que la homografía es casi innecesaria en el experimento y basta un *T* de menos de 5% para lograr precisiones cercanas al 100%.

Sin embargo imponer restricciones de color hace que muchos calces queden fuera en primera instancia, dándole al clasificador un sesgo de color que hace que ciertas instancias no sean recuperadas incluso en las condiciones menos restrictivas.

Tabla 4.6 Desempeño SIFT color transformado

Template	*Precision*	*Recall*	*F-Score*	*Template*	*Precision*	*Recall*	*F-Score*
Ahumada	0.98	0.98	0.98	Movistar	0.48	0.66	0.55
Cruz Verde	0.96	0.97	0.96	Ripley	0.94	0.95	0.94
Derco	0.99	0.99	0.99	CDF	0.69	0.79	0.74
Easy	0.74	0.96	0.84	Paris	0.87	0.98	0.92
Lanpass	0.92	0.90	0.91	Jumbo	0.84	0.99	0.90
Mitsubishi	0.96	0.97	0.97	Club LT	0.65	0.84	0.73
Sky	0.99	0.99	0.99	Nissan	0.92	0.98	0.95
Agrupemonos	0.99	0.99	0.99	Claro	0.51	0.36	0.42
Eclass	0.99	0.99	0.99	Samsung	0.87	0.90	0.88
Entel	0.74	0.91	0.82	Twitter Logo	0.40	0.71	0.52
Falabella	0.37	0.37	0.37	Twitter Tipo	0.28	0.21	0.24
Fiat	0.83	0.85	0.84	La tercera	0.39	0.46	0.42
Salcobrand	0.88	0.93	0.91	Facebook	0.29	0.39	0.33
Unimarc	0.33	0.44	0.37	**Promedio**	0.73	0.79	0.76

4.2.7 Experimento 6: *Opponent*-SIFT + MLSAC



a) b)

Ilustración 4.9 *Oponent* SIFT + MLSAC.

SIFT Oponente es según Van de Sande una solución efectiva para integrar información de color a **SIFT**, logrando el mejor desempeño dentro de las estrategias evaluadas en [15].

La solución consiste en agregar el **histograma oponente** como parte del descriptor SIFT. Al usar este descriptor de largo 384 se logra un resultado similar a la estrategia anterior, aunque debido a la implementación, **3 clases quedaron fuera** de la pirámide de escalas y por lo tanto **no fueron detectados**.(filas en negrita en Tabla 4.7).

Más allá de ese problema el desempeño del clasificador es usualmente la **mejor estrategia para la mayoría de los logos** de estos experimentos, integrar el histograma entrega una solución más flexible que SIFT de color transformado, aunque de todas maneras cerca del 20% de las instancias no son recuperadas bajo ninguna configuración.

Los resultados se pueden observar en la Ilustración 4.9.

Tabla 4.7 Desempeño SIFT Oponente

Template	*Precision*	*Recall*	*F-Score*	*Template*	*Precision*	*Recall*	*F-Score*
Ahumada	1.00	0.98	0.99	Movistar	1.00	0.38	0.55
Cruz Verde	1.00	0.97	0.98	Ripley	0.98	0.86	0.92
Derco	0.99	0.87	0.92	CDF	0.86	0.63	0.73
Easy	0.94	0.51	0.66	Paris	1.00	0.84	0.91
Lanpass	0.92	0.96	0.94	Jumbo	0.99	0.86	0.92
Mitsubishi	1.00	0.32	0.48	Club LT	0.94	0.64	0.76
Sky	1.00	0.99	0.99	Nissan	0.99	0.81	0.89
Agrupemonos	1.00	0.98	0.99	Claro	0.97	0.83	0.90
Eclass	1.00	0.98	0.99	Samsung	0.86	0.79	0.82
Entel	0.90	0.44	0.59	**Twitter Logo**	**1.00**	**0.00**	**0.00**
Falabella	1.00	0.33	0.50	**Twitter Tipo**	**1.00**	**0.00**	**0.00**
Fiat	1.00	0.39	0.56	La tercera	0.85	0.67	0.75
Salcobrand	1.00	0.99	0.99	**Facebook**	**1.00**	**0.06**	**0.11**
Unimarc	1.00	0.48	0.65	**Promedio**	0.73	0.79	0.76

Experimento 7: Viola Jones *detection framework*

Para evaluar el *framework* de detección Viola-Jones primero se separó la base de datos en una base de entrenamiento y validación, para así poder entrenar cada uno de las 27 cascadas de clasificadores:

Tabla 4.8 : Separacion de bases de datos

Base entrenamiento	60% instancias por logo
Base Validación	40 % instancias por logo
Base Negativos	300 paginas sin ningún logo. La misma base para los 27 entrenamientos.

Ilustración 4.10: Conjuntos de entrenamiento Viola-Jones

El clasificador logra un desempeño poco preciso y exhaustivo para la mayoría de las clases, probablemente debido a que las bases de entrenamiento no son suficientemente diversas para entrenar el clasificador.

Sin embargo, justo para aquellas figuras simples que presentan problemas en la implementación de descriptores locales, el clasificador logra mucho mejor desempeño probablemente debido a que se requieren menos etapas para clasificar una figura más simple, y por lo tanto menos templates para realizar el entrenamiento.

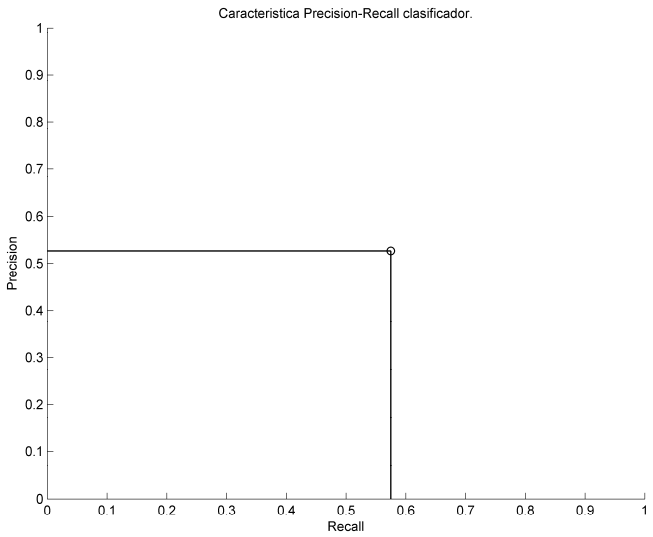


Ilustración 4.11 Desempeño Viola Jones, solo una configuración.

Tabla 4.9 Desempeño Viola Jones

Template	*Precisión*	*Recall*	*F-Score*	*Template*	*Precisión*	*Recall*	*F-Score*
Ahumada	0.27	0.15	0.19	Movistar	0.73	1.00	0.84
Cruz Verde	0.28	0.18	0.22	Ripley	0.08	0.08	0.08
Derco	0.63	0.89	0.74	CDF	0.70	0.98	0.82
Easy	0.40	0.19	0.26	Paris	0.40	0.60	0.48
Lanpass	0.63	0.88	0.73	Jumbo	0.55	0.21	0.30
Mitsubishi	0.53	0.60	0.56	Club LT	0.42	0.36	0.39
Sky	0.67	0.97	0.80	Nissan	0.81	0.75	0.78
Agrupemonos	0.58	0.98	0.73	Claro	0.52	0.34	0.41
Eclass	0.22	0.15	0.18	Samsung	0.71	0.68	0.69
Entel	0.78	0.89	0.83	Twitter Logo	0.69	0.93	0.79
Falabella	0.22	0.15	0.18	Twitter Tipo	0.76	0.95	0.84
Fiat	0.07	0.05	0.06	La tercera	0.48	0.79	0.60
Salcobrand	0.28	0.18	0.22	Facebook	0.83	0.87	0.85
Unimarc	0.99	0.75	0.85	**Promedio**	0.53	0.58	0.55

4.2.8 Experimento 8: Viola Jones + Histograma

Para integrar **información de color** se agrega un último clasificador débil a la cascada, que consiste calcular un histograma de color de 3 bines por canal RGB, dicho clasificador es aplicado a la salida de Viola-Jones y discrimina usando distancia entre el histograma de la ventana del candidato, con el histograma del template[61].

El histograma de cada canal es normalizado y discretizados en 3 niveles, lo cual genera un vector de 9 bits se descartan los candidatos que tienen una distancia de hamming mayor a 1,

De esa manera es posible filtrar falsos positivos mediante el uso de color, lo cual aumenta la precisión del clasificador, y disminuye levemente el *recall*.

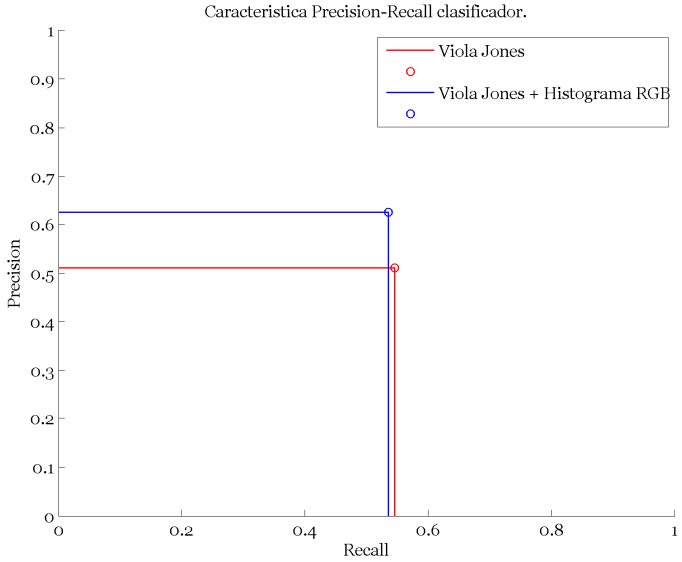


Ilustración 4.12: Precisión-Recall Viola-Jones + Histograma

Tabla 4.10 Desempeño Viola Jones + Histograma

Template	*Precisión*	*Recall*	*F-Score*	*Template*	*Precisión*	*Recall*	*F-Score*
Ahumada	0.43	0.15	0.22	Movistar	0.85	1.00	0.92
Cruz Verde	0.33	0.18	0.23	Ripley	0.27	0.08	0.12
Derco	0.87	0.89	0.88	CDF	0.70	0.98	0.82
Easy	0.51	0.19	0.28	Paris	0.40	0.60	0.48
Lanpass	0.68	0.88	0.77	Jumbo	0.55	0.21	0.30
Mitsubishi	0.58	0.60	0.59	Club LT	0.83	0.36	0.51
Sky	0.72	0.97	0.83	Nissan	0.83	0.75	0.79
Agrupemonos	0.88	0.98	0.93	Claro	0.52	0.34	0.41
Eclass	0.54	0.15	0.23	Samsung	0.75	0.62	0.68
Entel	0.85	0.89	0.87	Twitter Logo	0.83	0.93	0.88
Falabella	0.31	0.15	0.20	Twitter Tipo	0.86	0.90	0.88
Fiat	0.16	0.05	0.08	La tercera	0.63	0.70	0.66
Salcobrand	0.53	0.18	0.26	Facebook	0.91	0.76	0.83
Unimarc	0.99	0.75	0.85	**Promedio**	0.64	0.56	0.60

[61] Se considera el mismo template que para generar los descriptores.

4.2.9 Todas las estrategias:

En la Tabla 4.11 es posible observar el *F-Score* que logran las distintas estrategias, colores verdes representan buenos desempeños, mientras que los rojos son malos desempeños.

Todos los gráficos precisión/recall se encuentran en el anexo 7.2.

Tabla 4.11 Desempeño todas las estrategias.

	F-Score							
Logo	**Template Matching**	**Freak**	***Csift***	**Sift**	**Surf**	**Viola Jones**	**Viola Jones + Histograma**	**Sift Oponente**
'AHUMADA'	0.04	0.89	0.99	0.98	0.97	0.19	0.22	1.00
'CRUZVERDE'	0.03	0.98	0.99	0.96	0.99	0.22	0.23	0.99
'DERCO'	0.05	0.95	0.95	0.99	0.99	0.74	0.88	0.99
'EASY'	0.03	0.49	0.78	0.84	0.44	0.26	0.28	0.95
'LANPASS'	0.04	0.77	0.93	0.91	0.98	0.73	0.77	0.98
'MITSUBISHI'	0.11	0.82	0.65	0.97	0.98	0.56	0.59	0.88
'SKY'	0.02	0.96	1.00	0.99	1.00	0.80	0.83	1.00
'AGRUPEMONOS'	0.03	0.98	0.99	0.99	0.99	0.73	0.93	0.99
'ECLASS'	0.07	0.99	0.99	0.99	0.99	0.18	0.23	0.99
'ENTEL'	0.04	0.58	0.71	0.82	0.80	0.83	0.87	0.70
'FALABELLAL'	0.09	0.79	0.67	0.37	0.84	0.18	0.20	0.92
'FIAT'	0.03	0.74	0.72	0.84	0.80	0.06	0.08	0.62
'SALCOBRAND'	0.07	0.63	1.00	0.91	0.94	0.22	0.26	1.00
'UNIMARC'	0.03	0.79	0.79	0.37	0.61	0.85	0.85	0.95
'MOVISTAR'	0.06	0.55	0.71	0.55	0.68	0.84	0.92	0.68
'RIPLEY'	0.02	0.70	0.94	0.94	0.96	0.08	0.12	0.96
'CDF'	0.04	0.45	0.79	0.74	0.23	0.82	0.82	0.76
'PARIS'	0.03	0.79	0.95	0.92	0.79	0.48	0.48	0.97
'JUMBO'	0.02	0.58	0.95	0.90	0.74	0.30	0.30	0.95
'CLUBLATERCERA'	0.02	0.68	0.84	0.73	0.90	0.39	0.51	0.86
'NISSAN'	0.02	0.90	0.94	0.95	0.96	0.78	0.79	0.98
'CLARO'	0.09	0.88	0.93	0.42	0.95	0.41	0.41	0.82
'SAMSUNG'	0.08	0.77	0.84	0.88	0.88	0.69	0.68	0.87
'LTWITTER'	0.00	0.28	0.00	0.52	0.16	0.79	0.88	0.00
'TWITTERT'	0.10	0.05	0.00	0.24	0.29	0.84	0.88	0.02
'LATERCERA'	0.08	0.32	0.80	0.42	0.60	0.60	0.66	0.72
'FACEBOOK'	0.00	0.06	0.20	0.00	0.20	0.85	0.83	0.41
Promedio	0.05	0.71	0.83	0.76	0.78	0.55	0.60	0.85

4.3 Análisis de los resultados

4.3.1 Mejor estrategia por template

En la sección anterior se presentaron los resultados de los experimentos realizados en detección de logos, si bien algunas estrategias resultaron mejores que otras, todas tienen falencias como clasificadores ante ciertos *templates*.

En la Tabla 4.12 se puede ver que eligiendo la mejor estrategia para cada *template* se puede lograr un desempeño de:

Precisión	0.91
Recall	0.92
F-Score	0.91

Tabla 4.12 Mejor resultado por template, en verde templates 'simples'.

Template	*Estrategia*	*Precision*	*Recall*	*F-Score*	*Template*	*Estrategia*	*Precision*	*Recall*	*F-Score*
Ahumada	OPONENTE	1.00	0.98	0.99	Movistar	VJH	0.85	1.00	0.92
Cruz Verde	SURF	0.98	1.00	0.99	Ripley	SURF	0.93	1.00	0.96
Derco	SURF	0.99	1.00	0.99	CDF	VJH	0.70	0.98	0.82
Easy	SIFT	0.74	0.96	0.84	Paris	OPONENTE	1.00	0.84	0.91
Lanpass	SURF	0.98	0.98	0.98	Jumbo	OPONENTE	0.99	0.86	0.92
Mitsubishi	SURF	0.95	1.00	0.98	Club LT	SURF	0.88	0.92	0.90
Sky	SURF	0.99	1.00	1.00	Nissan	SURF	0.92	0.99	0.96
Agrupemonos	SURF	0.99	1.00	0.99	Claro	SURF	0.94	0.97	0.95
Eclass	SURF	0.99	1.00	0.99	Samsung	OPONENTE	0.86	0.79	0.82
Entel	VJH	0.85	0.89	0.87	Twitter Logo	VJH	0.83	0.93	0.88
Falabella	SURF	0.86	0.82	0.84	Twitter Tipo	VJH	0.86	0.90	0.88
Fiat	SIFT	0.83	0.85	0.84	La tercera	OPONENTE	0.85	0.67	0.75
Salcobrand	OPONENTE	1.00	0.99	0.99	Facebook	VJH	0.83	0.87	0.85
Unimarc	VJH	0.99	0.75	0.85	**Promedio**	Mixto	0.91	0.92	0.91

Este margen se da principalmente porque las estrategias basadas en descriptores tienen un mal desempeño en un conjunto reducido de logos 'simples', aquellos que mientras que estos tienen un buen desempeño al aplicar Viola Jones + Histograma RGB, como se muestra en la Ilustración 4.13.

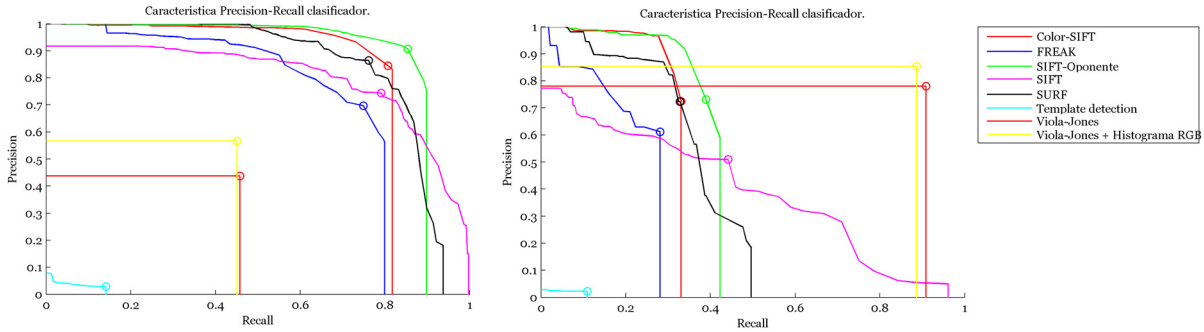


Ilustración 4.13 Desempeño templates 'complejos', a la izquierda, vs 'simples' a la derecha.

En la Ilustración 4.14 se observa el desempeño de distintas estrategias:

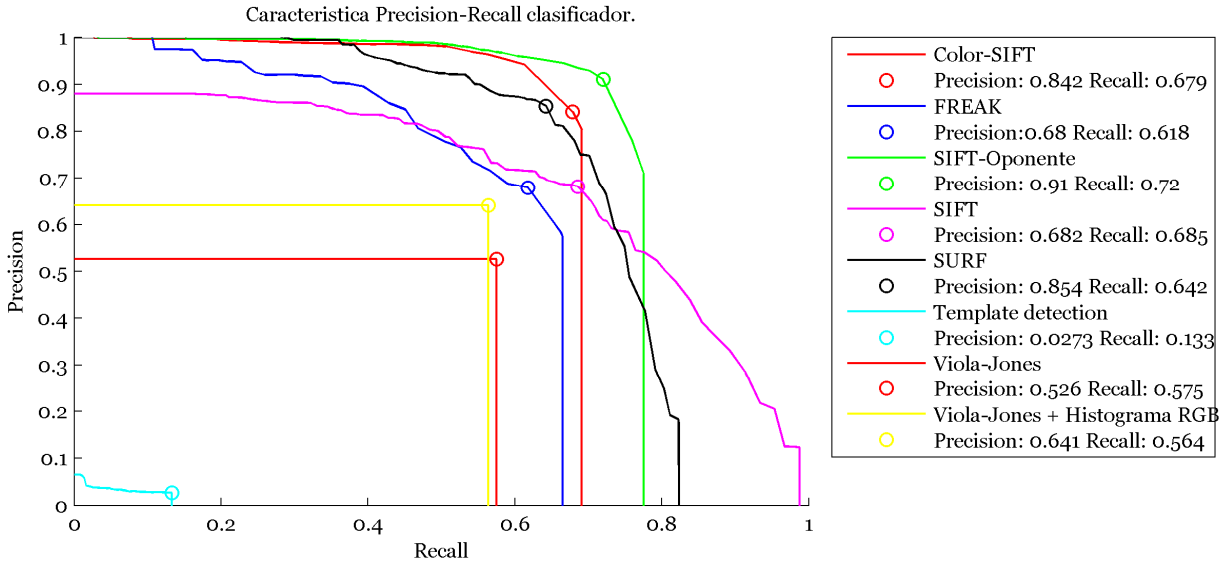


Ilustración 4.14 Desempeño de todas las estrategias

Y el desempeño tomando la mejor estrategia para cada template en la Ilustración 4.15:

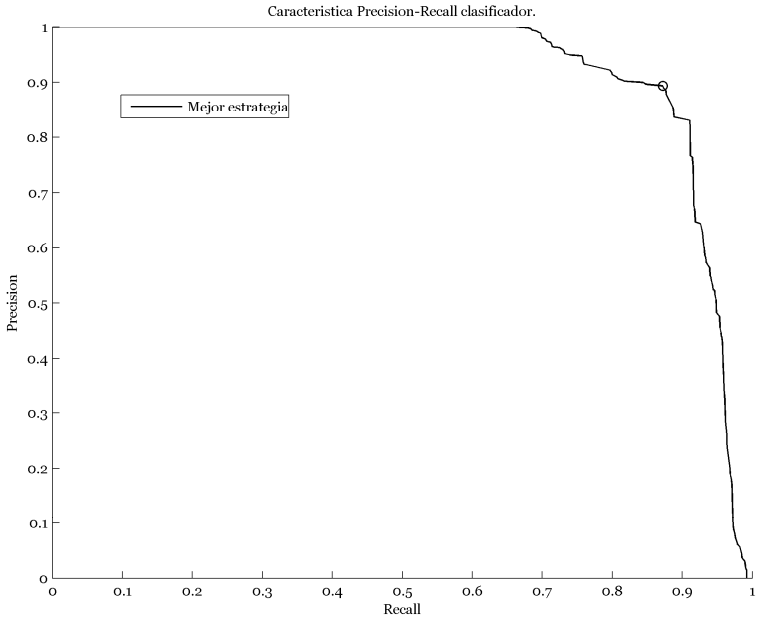


Ilustración 4.15: Desempeño mejor estrategia por logo.

4.3.2 Análisis de errores.

Para analizar el desempeño en las estrategias es necesario analizar dos fenómenos:

- Donde y porque aparecen **falsos positivos**.
- Donde y porque el detector no detecta los logos, **falsos negativos**.

4.3.2.1. Errores relativos a la definición de las clases.

Al analizar los calces, tanto para estrategias de descriptores locales como para Viola Jones, se encuentra un problema estructural con el planteamiento del problema, se definen las clases como logos gráficamente similares, como en la Ilustración 4.16

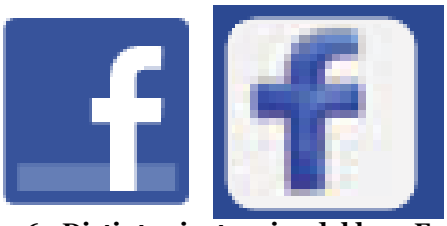


Ilustración 4.16 : Distintas instancias del logo Facebook.

Si bien el logo es similar a la vista del ojo humano y ambos corresponden a una 'f' rodeada por un rectángulo, desde el punto de vista de intensidades uno es similar a la negación del otro.

Las implicancias de problema son:

- Los descriptores basados en gradiente están invertidos, esto significa que al calcular los descriptores de $I_{template}$con I_{sample} , no solo son diferentes sino que el gradiente es el opuesto. Eso hace que los descriptores sean particularmente distantes y no haya posibilidad alguna de que existan calce.

- Al entrenar Viola-Jones, se está entregando información no coherente en el entrenamiento, al no existir coherencia entre las imágenes, es muy difícil que el entrenamiento de clasificadores entregue una salida correcta cuando el entrenamiento es la negación de la clase.

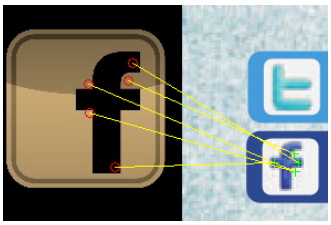


Ilustración 4.17: Calces (no validados) con la imagen invertida, la imagen sin invertir no genera calces.

El problema puede ser solucionado generando una nueva clase con el logo invertido en el caso de descriptores locales, y en el caso de Viola-Jones se debe entrenar como dos clasificadores distintos.

De todas maneras la definición de clases se identifica como una de las principales fuente de falsos negativos, es decir, es esperable que el detector no detecte gráficas que estrictamente no son las mismas, por ese motivo, es muy difícil esperar tener *recall* de 100%, ya que la definición de clases es ambigua.

También existen clases que se solapan (Ilustración 4.18), ya que tienen zonas similares, por ejemplo en la siguiente figura se observa como existe ambigüedad entre logos que comparten la palabra 'cencosud' dentro.

Ilustración 4.18: Logos tienen puntos de interés que se solapan, SURF.

Entre las dos imágenes existe una homografía valida, por lo que el problema recae en la definición de las clases, más que a limitaciones de la estrategia usada. Este problema es eliminado usando estrategias de color, como se ve en la Ilustración 4.19.

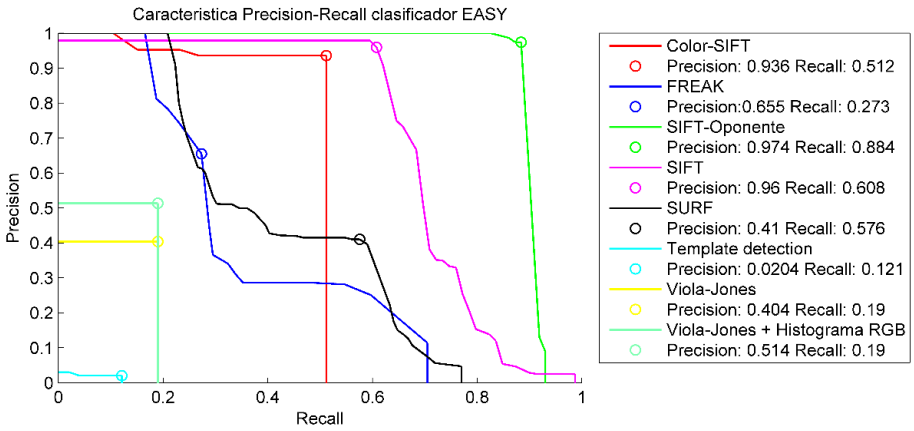


Ilustración 4.19 Desempeño de distintas estrategias para el clasificador 'EASY'

También como generalmente los logos contienen elementos tipográficos, existen instancias donde estos son calzados con la tipografía del diario, aunque estos errores deben ser evitados aumentando la exigencia del clasificador, el umbral T de la homografía.

Ilustración 4.20: Error, SIFT Oponente.

4.3.2.2. Errores relativos a la implementación.

En el **experimento 2** se observa una gran tasa de falsos positivos, esto se debe a que el cálculo de la homografía acepta homografías proyectivas de todo tipo:

Ilustración 4.21 : Homografías incoherentes, falso positivo.

Viola-Jones

La cascada de filtros *Haar-like* construida por la estrategia Viola-Jones, presenta un desempeño mediocre en los experimentos 7 y 8, esto se debe a que la base de entrenamiento es muy pequeña y por lo tanto la cascada considera características muy básicas.

Entonces el detector mal entrenado considera ciertas características muy simples, pero no logra conseguir una precisión ni *recall* esperado. En la Ilustración 4.22 se puede ver

como el clasificador entrenado para detectar el logo 'salcobrand' detecta círculos, generando muchos falsos positivos.

a) Vp b) Fp c) Fp d) Fp

Ilustración 4.22 : Ejemplos verdaderos positivos y falsos positivos Viola-Jones, clasificador 'salcobrand'.

Este problema es resuelto parcialmente al descartar en una última etapa usando el histograma de color, calculado sobre el candidato y comparado con el histograma original del template. Esto corrige levemente la ***precisión*** del clasificador, pero para la mayoría de los templates el ***recall*** sigue siendo muy bajo comparado con el desempeño de otras estrategias.

4.3.2.3. Errores debidos a distorsiones.

Las principales fuentes de distorsiones son:

- Traslaciones
- Escalamientos.
- Oclusiones
- Rotaciones.
- Compresión.

Al inspeccionar los **falsos positivos** y **negativos** de los experimentos, sin embargo se observa que son los logos escalados aquellos que presentan mayor cantidad de falsos negativos, es decir, no son recuperados.

Repetitividad en escalamiento

El análisis de repetitividad consiste en evaluar los calces de la imagen consigo mismo luego de distorsionarla, en este caso se evalúa la repetitividad de los templates con respecto a escalamientos.

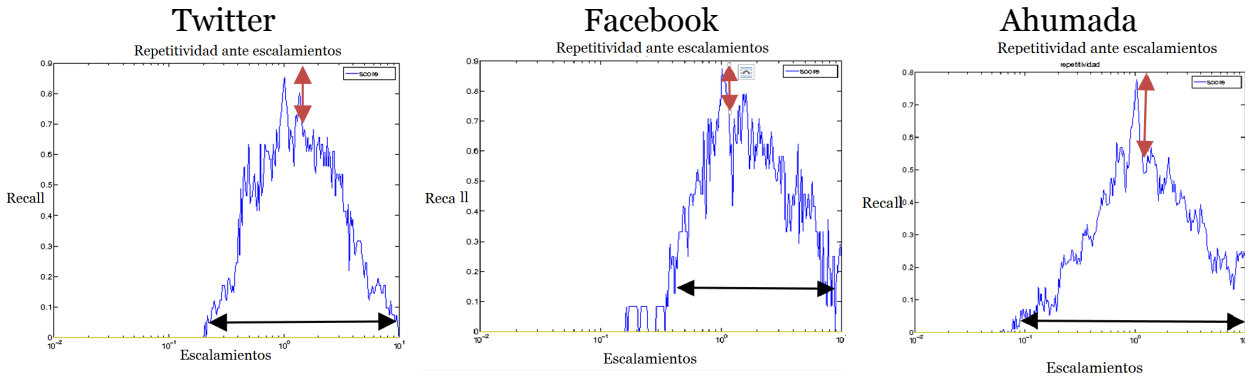
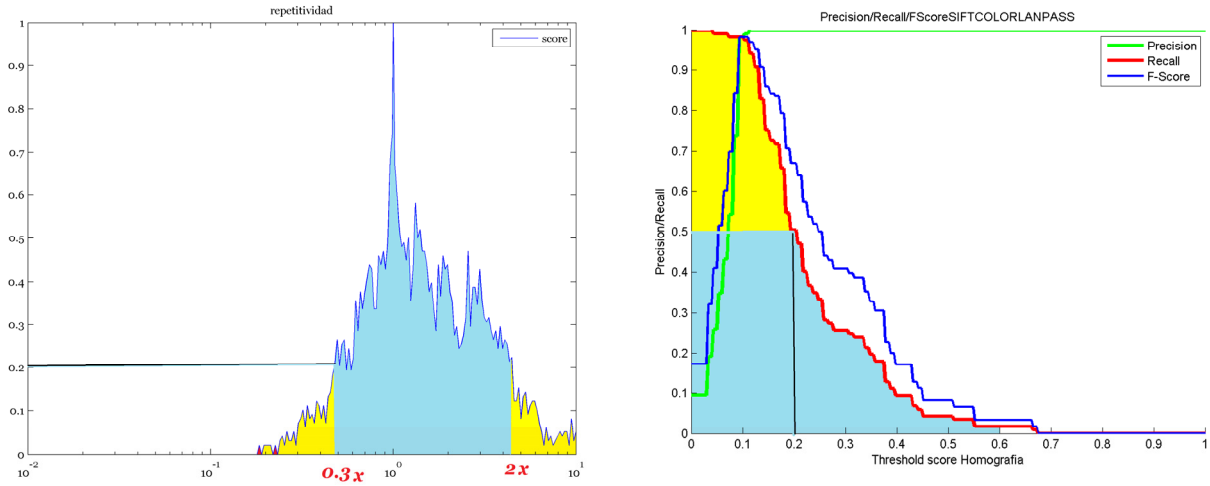


Ilustración 4.23 Repetitividad en escala, SURF

Logos más complejos como 'ahumada' presentan un *peak* de características *'volátiles'* que se pierden inmediatamente al escalar la imagen, representado por la flecha roja en la Ilustración 4.23, por otro lado se puede observar que si bien las características se pierden rápidamente, existen ciertas características que presentan *resiliencia* y siguen persistiendo ante el escalamiento de la imagen, identificados por la flecha negra en la Ilustración 4.23.

La **repetitividad** esta intrínsecamente relacionada con el ***recall*** del clasificador.



a) b)

Ilustración 4.24 Relación entre repetitividad y recall, ejemplificado con LANPASS y SURF.

Cuando se fija un *threshold* T, implícitamente existe un sesgo de escala, en la Ilustración 4.24 se observa como al fijar $T = 0.2$, implícitamente se esta rechazando las instancias del logo 'lanpass' con escalas menores a 0.3x y mayores a 2x, en las figuras a) y b) se representan dichas instancias como el área amarilla, mientras que el área celeste son las instancias recuperadas.

El escalamiento no es la única distorsión que tiene la imagen, existen otros efectos por los que se recupera solo el 50% de las instancias, sin embargo la mayoría de los falsos negativos en el sistema son justamente casos como el ilustrado en Ilustración 4.25 , donde el logo b) está en una escala irrecuperable para el clasificador. Cabe destacar que este caso, corresponde a dimensiones menores a $1cm^2$, que es el requerimiento del sistema presentado en el capítulo 3.

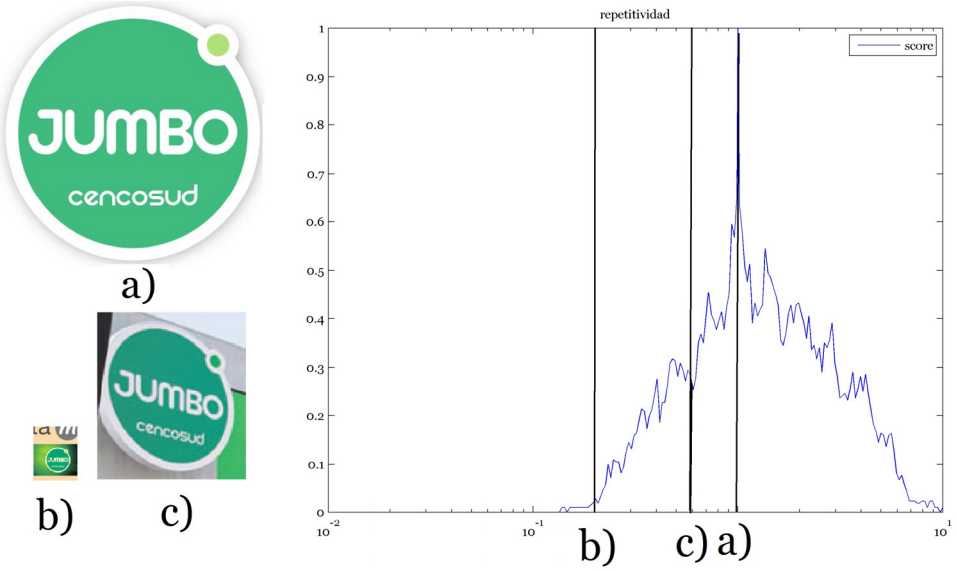


Ilustración 4.25: Logo b), irrecuperables en escala.

4.3.2.4. Estrategias para disminuir el error

Las fuentes de error mencionadas anteriormente pueden ser aplacadas mediante el uso de:

- Agregar *templates*: Usando *templates* invertidos, o usando varias representaciones graficas de un logo es posible disminuir falsos negativos provenientes de las grandes diferencias intra-clase.

- Agregar templates en distintas escalas: Usando templates en distintas escalas es posible recuperar logos en escalas muy pequeñas, que actualmente son irrecuperables en escala, como se puede observar en la Ilustración 4.25.

- Rechazar homografías poco verosímiles: Usando restricciones sobre la matriz de homografía es posible refinar la clasificación.

- Trabajar en alta resolución, evitando distorsiones por muestreo: Al trabajar en alta resolución es posible recuperar puntos de interés que no serían considerados en resoluciones bajas.

Capítulo 5 Conclusiones:

En este trabajo se estudia el problema de detección de objetos, aplicado al problema de detección de publicidad en prensa escrita, para lo cual se presenta una estrategia de extracción de información publicitaria relevante, basado principalmente en el cálculo de descriptores locales de la imagen. También se construyó una base de datos de testeo acorde al problema.

El sistema incluye una etapa de preprocesamiento, que mediante el uso de operaciones simples de erosión y dilación logra efectivamente eliminar regiones de texto dentro de la imagen. Lo cual permite reducir el número de puntos de interés en cerca de un 70%.

Luego se aplica una etapa de detección de logos, que permite detectar logos con precisiones en el rango del 90-100% de precisión y *recall*, como se observa en los experimentos realizados sobre la base de datos.

La tercera etapa consiste en hacer calce de anuncios publicitarios candidatos dentro de la imagen, etapa que puede presentar desempeños cercanos al 100%, debido al buen desempeño de estrategias de descriptores locales para calzar imágenes complejas.

Finalmente basada en los parámetros de tarificación del diario y la estimación de la sección del anuncio, se puede obtener un precio estimado del anuncio.

Los experimentos realizados sobre detección de logos muestran que el uso de descriptores locales es una herramienta poderosa y flexible para la detección de logos, sea en sus variantes SURF, SIFT, FREAK, CSIFT o SIFT-Oponente, aunque en ciertos logos simples se observa que el desempeño suele decaer.

Por otro lado dichos logos simples pueden ser fácilmente reconocidos usando un clasificador entrenado, como es el uso de la estrategia *Viola-Jones detection framework*.

Todas las estrategias pueden ser acompañadas de información de color que permite agregar una nueva dimensión a la clasificación.

Se hace un análisis de los resultados, donde se identifican los problemas que generan clasificaciones erróneas dentro de la base de datos de validación. Se identifican errores ocasionados por la definición de clases no equivalentes, que no generan descriptores similares. También se identifican errores debido a que los descriptores de elementos tipográficos se solapan con elementos de la imagen, a homografías validadas incorrectamente y principalmente debido a distorsiones de escalamiento.

Al escoger la mejor estrategia para cada *template* se logra una precisión de 91% y un *recall* promedio de 90% en detección de logos, sobre la base de testeo, eso sumado con los resultados expuestos en detección de anuncios y detección de encabezados muestra que el proyecto *IntelliMEDIA* puede ser implementado como una solución práctica para la detección de anuncios publicitarios en prensa escrita.

Capítulo 6 Bibliografía

[1] LCHV. [En línea]. Disponible en: http://www.logoschilevector.cl/.

[2] L. tercera, «Tarifario,» 2014. [En línea]. Disponible en: http://www.tarifaspublicitarias.com/

[3] Wikipedia, «Structuring_element,» [En línea]. Disponible en: http://en.wikipedia.org/wiki/Structuring_element.

[4] T. Lindeberg, «Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention,» *International Journal of Computer Vision,* vol. 11, nº 3, pp. 283--318, 1993.

[5] Wikipedia, Pattern recognition [En línea]. Disponible en: http://en.wikipedia.org/wiki/Pattern_recognition.

[6] P. H. a. Z. A. Torr, «MLESAC: A new robust estimator with application to estimating image geometry,» *Computer Vision and Image Understanding,* vol. 78, nº 1, pp. 138--156, 2000.

[7] Y. a. S. R. E. Freund, «A decision-theoretic generalization of on-line learning and an application to boosting,» *Journal of computer and system sciences,* vol. 55, nº 1, pp. 119-139, 1997.

[8] D. G. Lowe, «Distinctive image features from scale-invariant keypoints,» *International journal of computer vision,* vol. 60, nº 2, pp. 91-110, 2004.

[9] H. a. T. T. a. V. G. L. Bay, «Surf: Speeded up robust features,» de *Computer Vision--ECCV 2006*, Springer, 2006, pp. 404--417.

[10] M. a. L. V. a. S. C. a. F. P. Calonder, «Brief: Binary robust independent elementary features,» de *Computer Vision--ECCV 2010*, Springer, 2010, pp. 778--792.

[11] S. a. C. M. a. S. R. Y. Leutenegger, «BRISK: Binary robust invariant scalable keypoints,» de *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, 2011, pp. 2548--2555.

[12] E. a. R. V. a. K. K. a. B. G. Rublee, «ORB: an efficient alternative to SIFT or SURF,» de *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011.

[13] A. a. O. R. a. V. P. Alahi, «Freak: Fast retina keypoint,» de *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, pp. 510--517.

[14] E. a. R. V. a. K. K. a. B. G. Rublee, «ORB: an efficient alternative to SIFT or SURF,» de *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, 2011, pp. 2564--2571.

[15] K. E. a. G. T. a. S. C. G. Van De Sande, «Evaluating color descriptors for object and scene recognition,» *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 32, nº 9, pp. 1582--1596, 2010.

[16] J. a. G. T. a. B. A. D. Van De Weijer, «Boosting color saliency in image feature detection,» *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 28, nº 1, pp. 150--156, 2006.

[17] A. a. Z. A. a. M. X. Bosch, «Scene classification using a hybrid generative/discriminative approach,» *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 30, pp. 712--727, 2008.

[18] J.-M. a. V. d. B. R. a. S. A. W. M. a. G. H. Geusebroek, «Color invariance,» *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 23, nº 12, pp. 1338--1350, 2001.

[19] *Papel digital Disponible en papel www.papeldigital.cl.*

[20] computer-vision-talks, «A BATTLE OF THREE DESCRIPTORS: SURF, FREAK AND BRISK,» [En línea]. Disponible en: http://computer-vision-talks.com/articles/2012-08-18-a-battle-of-three-descriptors-surf-freak-and-brisk/.

[21] G. &. D. D. Zhu, «Automatic document logo detection.,» de *Ninth International Conference on Document Analysis and Recognition*, 2007.

[22] G. Z. a. Y. Z. a. D. D. a. S. Jaeger, «Tobacco 800 database,» [En línea]. Disponible en: http://www.umiacs.umd.edu/~zhugy/tobacco800.html.

[23] P. a. J. M. Viola, «Rapid object detection using a boosted cascade of simple features,» de *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, IEEE, 2001, pp. 1--511.

[24] D. S. a. R. E. a. W. I. Doermann, «Logo recognition using geometric invariants,» de *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*, IEEE, 1993, pp. 894--897.

[25] *SimpleCV, disponible en www.simplecv.org.*

[26] R. Szeliski, Computer vision algorithms and applications, 2011.

[27] http://www.vlfeat.org/, *VLFEAT.*

[28] Wikipedia, «Media_monitoring,» [En línea]. Disponible en: http://en.wikipedia.org/wiki/Media_monitoring.

[29] Wikipedia, «Media monitoring service,» [En línea]. Disponible en: http://en.wikipedia.org/wiki/Media_monitoring_service.

[30] MathWorks, «detectSURFFeatures documentation,» [En línea]. Disponible en: http://www.mathworks.com/help/vision/ref/detectsurffeatures.html.

[31] Wikipedia, «HSL and HSV,» [En línea]. Disponible en: http://en.wikipedia.org/wiki/HSL_and_HSV.

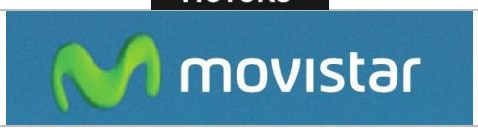
[32] MathWorks, «rgbhist,» [En línea]. Disponible en: http://www.mathworks.com/matlabcentral/fileexchange/43630-color-histogram-of-an-rgb-image/content/rgbhist.m.

[33] Charles K. Chui, "An Introduction to Wavelets," *Academic Press,* 1992.

Capítulo 7 Anexos

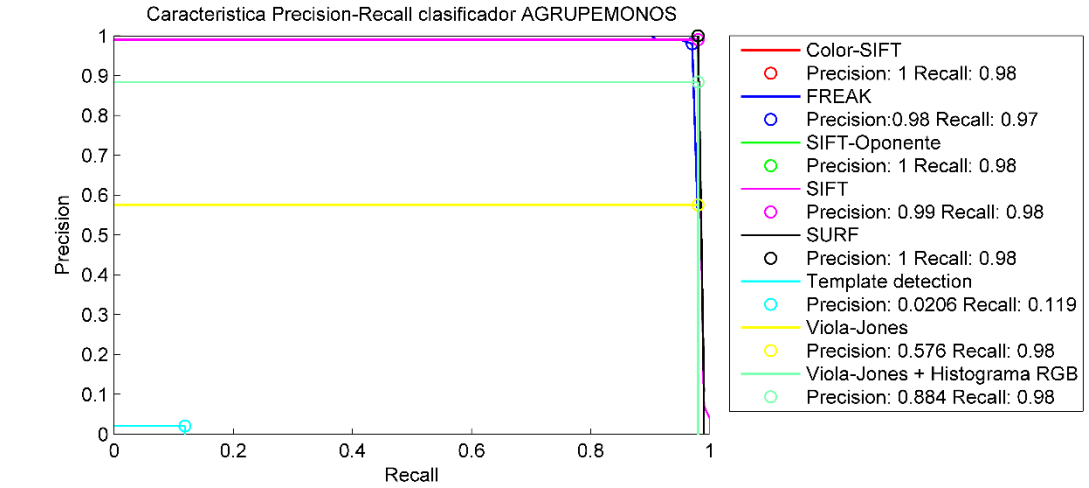
7.1 Anexo A: *Templates* usados.

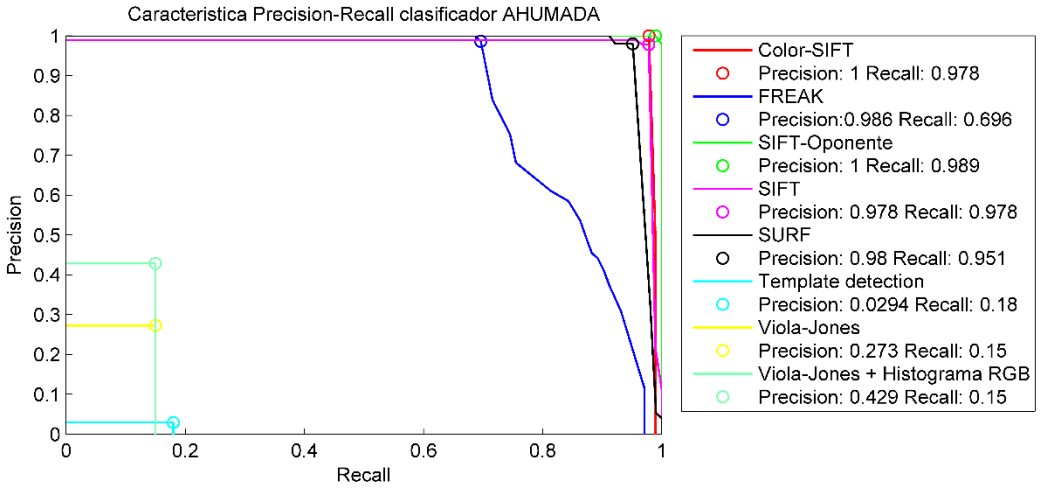
Agrupemonos	AGRUPEMONOS.CL
Ahumada	farmacias ahumada® TE CUIDA MEJOR SIN PAGAR DE MÁS
Claro	Claro
Club la tercera	Club LA TERCERA
Cruz verde	Farmacias Cruz Verde
Derco	DERCO
Easy	easy cencosud
E-class	eClass®

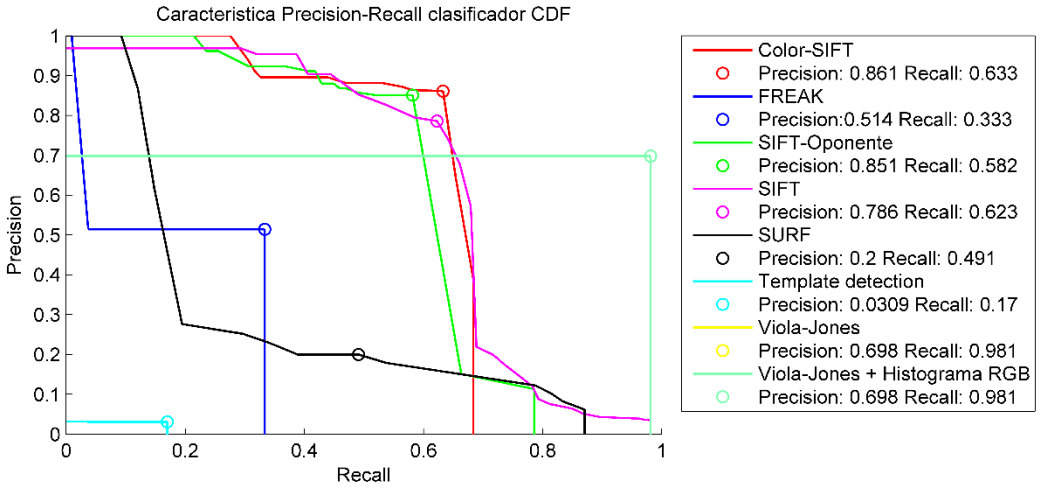
Entel	
Facebook	
Falabella	
Fiat	
Jumbo	
Lanpass	
La tercera	
Twitter gráfico	
Mitsubishi	
Movistar	

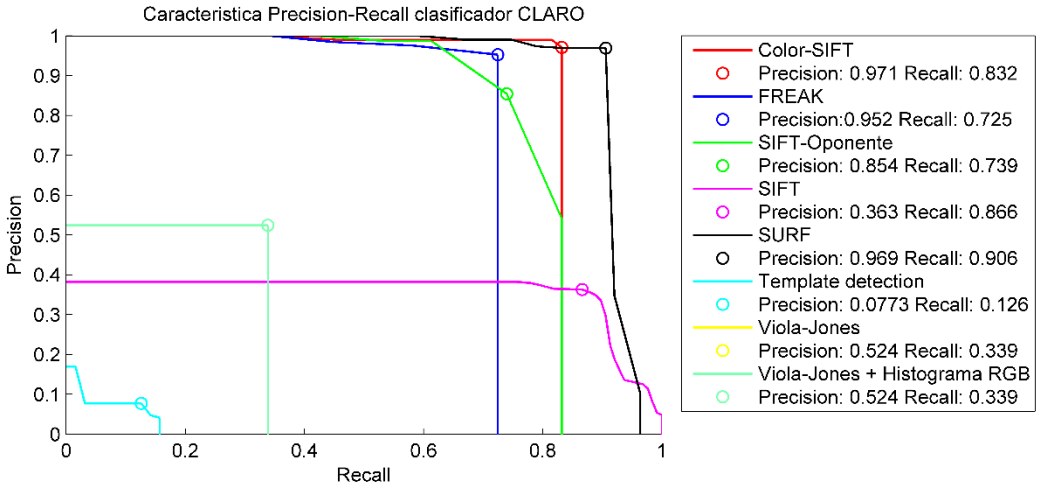
Nissan	
Paris	
Ripley	
Salcobrand	
Samsung	
Sky	
Twitter tipográfico	
Unimarc	
CDF	

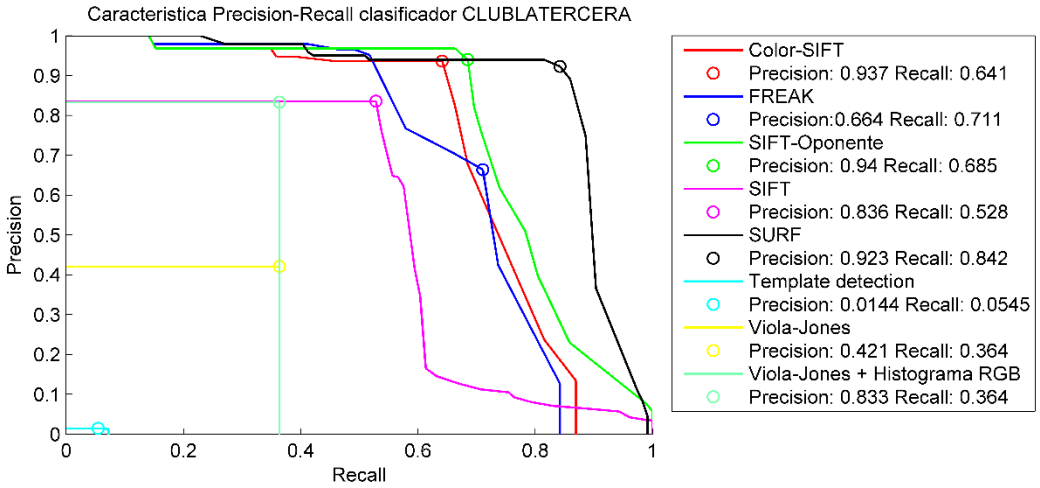
7.2 Anexo B: Curvas precisión-recall clasificadores por logo:

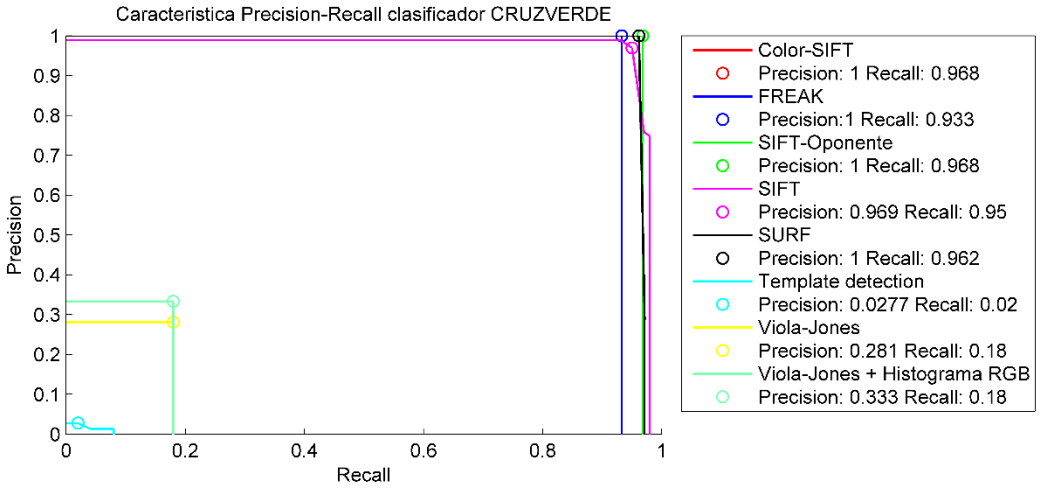


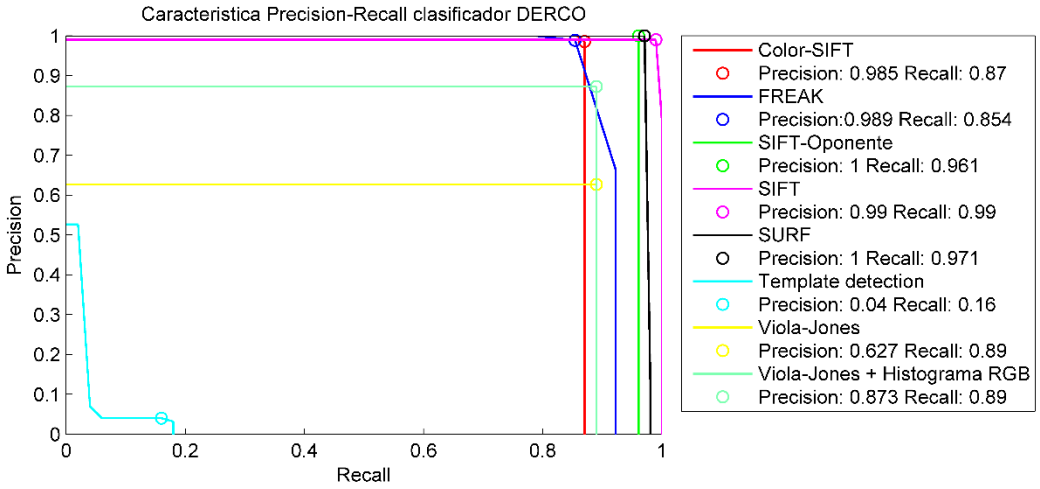


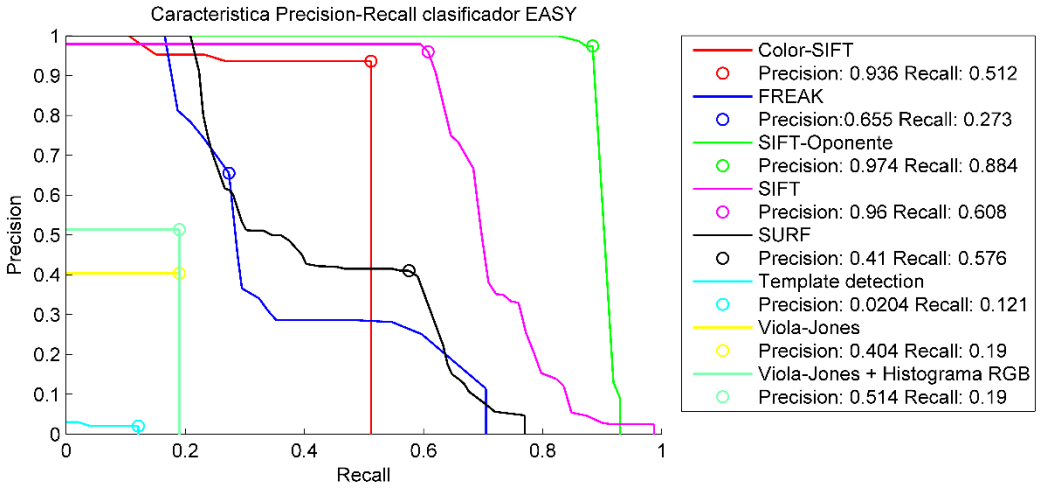


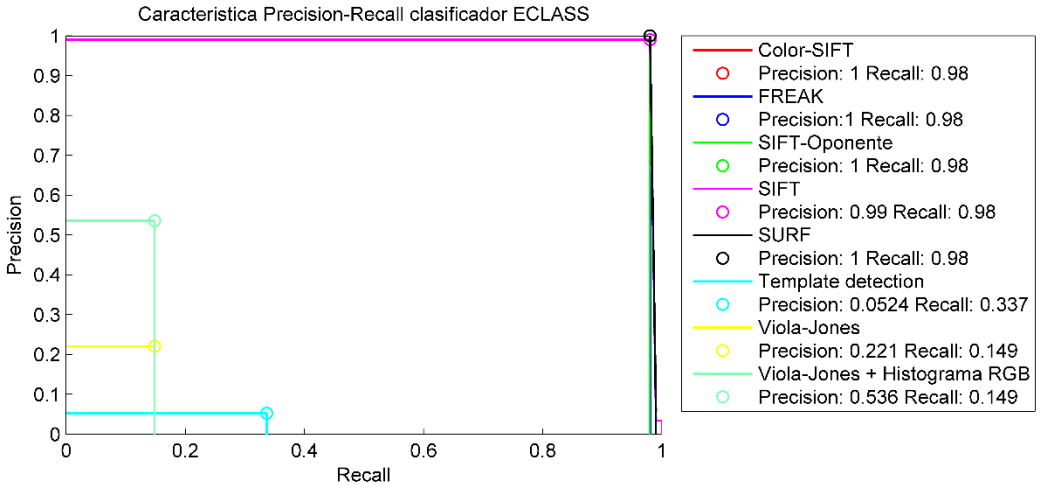

Caracteristica Precision-Recall clasificador CLARO
Precision
Recall
Color-SIFT
Precision: 0.971 Recall: 0.832
FREAK
Precision:0.952 Recall: 0.725
SIFT-Oponente
Precision: 0.854 Recall: 0.739
SIFT
Precision: 0.363 Recall: 0.866
SURF
Precision: 0.969 Recall: 0.906
Template detection
Precision: 0.0773 Recall: 0.126
Viola-Jones
Precision: 0.524 Recall: 0.339
Viola-Jones + Histograma RGB
Precision: 0.524 Recall: 0.339

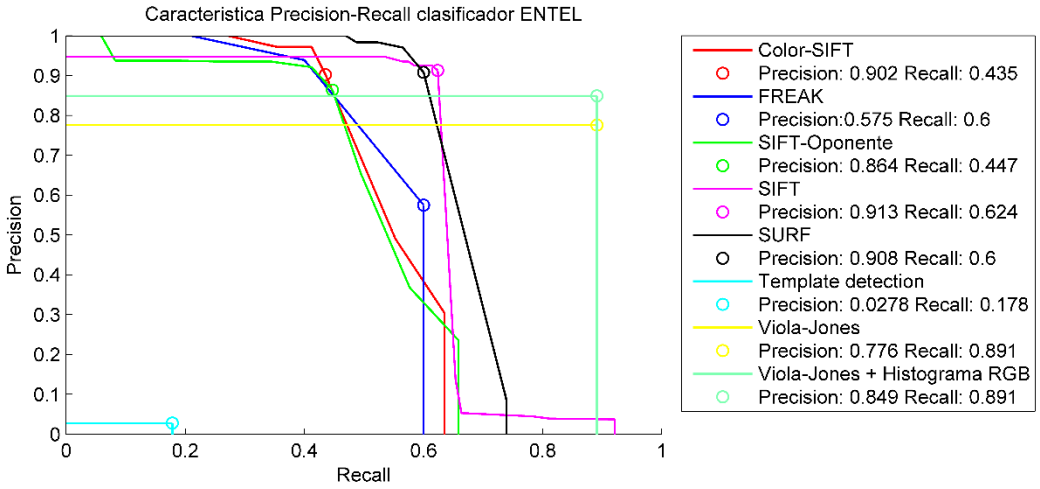

Caracteristica Precision-Recall clasificador CLUBLATERCERA
Precision
Recall
Color-SIFT
Precision: 0.937 Recall: 0.641
FREAK
Precision:0.664 Recall: 0.711
SIFT-Oponente
Precision: 0.94 Recall: 0.685
SIFT
Precision: 0.836 Recall: 0.528
SURF
Precision: 0.923 Recall: 0.842
Template detection
Precision: 0.0144 Recall: 0.0545
Viola-Jones
Precision: 0.421 Recall: 0.364
Viola-Jones + Histograma RGB
Precision: 0.833 Recall: 0.364

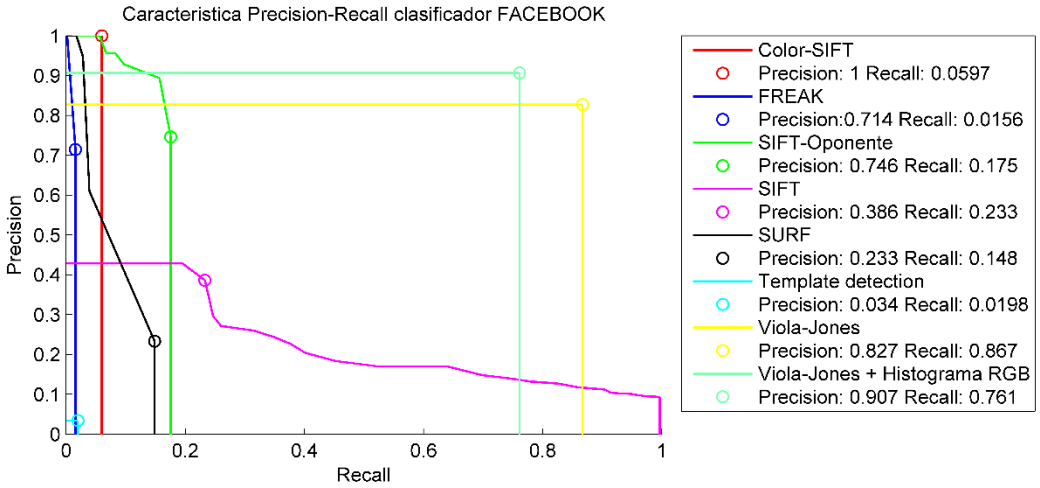

Caracteristica Precision-Recall clasificador CRUZVERDE
Precision
Recall
Color-SIFT
Precision: 1 Recall: 0.968
FREAK
Precision:1 Recall: 0.933
SIFT-Oponente
Precision: 1 Recall: 0.968
SIFT
Precision: 0.969 Recall: 0.95
SURF
Precision: 1 Recall: 0.962
Template detection
Precision: 0.0277 Recall: 0.02
Viola-Jones
Precision: 0.281 Recall: 0.18
Viola-Jones + Histograma RGB
Precision: 0.333 Recall: 0.18

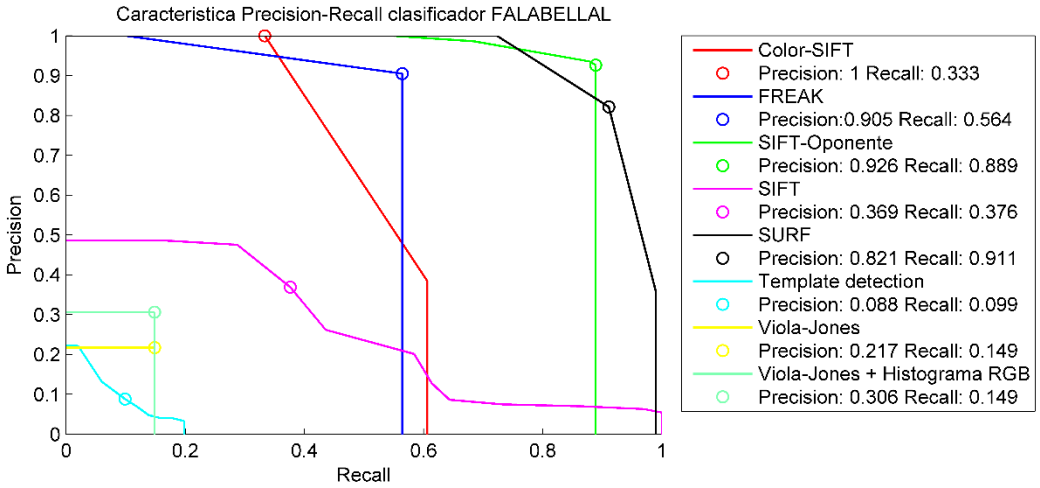

Caracteristica Precision-Recall clasificador DERCO
Precision
Recall
Color-SIFT
Precision: 0.985 Recall: 0.87
FREAK
Precision:0.989 Recall: 0.854
SIFT-Oponente
Precision: 1 Recall: 0.961
SIFT
Precision: 0.99 Recall: 0.99
SURF
Precision: 1 Recall: 0.971
Template detection
Precision: 0.04 Recall: 0.16
Viola-Jones
Precision: 0.627 Recall: 0.89
Viola-Jones + Histograma RGB
Precision: 0.873 Recall: 0.89

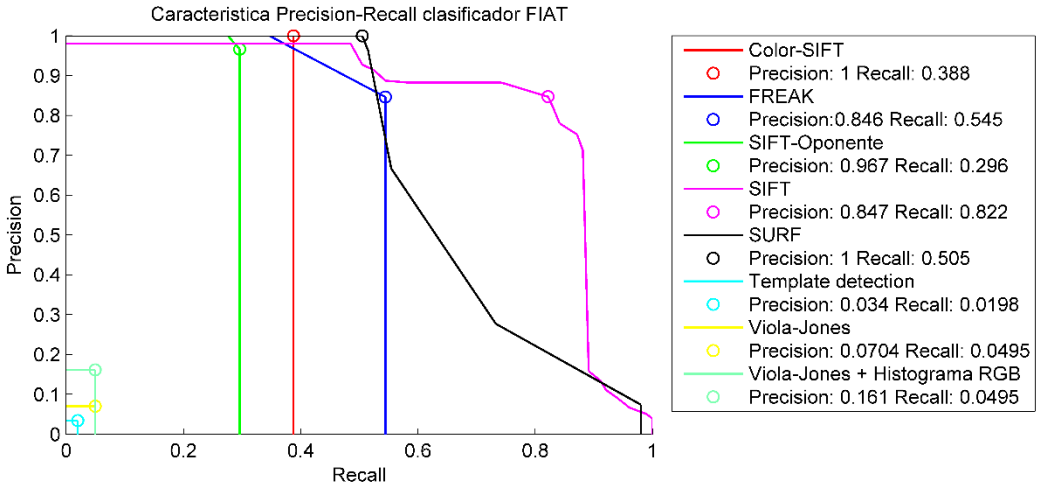

Caracteristica Precision-Recall clasificador EASY
Precision
Recall
Color-SIFT
Precision: 0.936 Recall: 0.512
FREAK
Precision:0.655 Recall: 0.273
SIFT-Oponente
Precision: 0.974 Recall: 0.884
SIFT
Precision: 0.96 Recall: 0.608
SURF
Precision: 0.41 Recall: 0.576
Template detection
Precision: 0.0204 Recall: 0.121
Viola-Jones
Precision: 0.404 Recall: 0.19
Viola-Jones + Histograma RGB
Precision: 0.514 Recall: 0.19

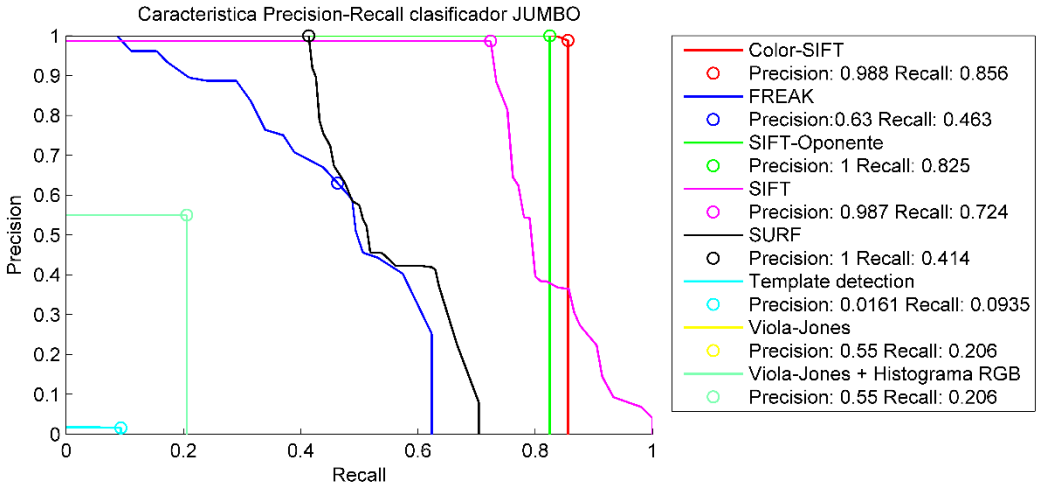

Caracteristica Precision-Recall clasificador ECLASS
Precision
Recall
Color-SIFT
Precision: 1 Recall: 0.98
FREAK
Precision:1 Recall: 0.98
SIFT-Oponente
Precision: 1 Recall: 0.98
SIFT
Precision: 0.99 Recall: 0.98
SURF
Precision: 1 Recall: 0.98
Template detection
Precision: 0.0524 Recall: 0.337
Viola-Jones
Precision: 0.221 Recall: 0.149
Viola-Jones + Histograma RGB
Precision: 0.536 Recall: 0.149

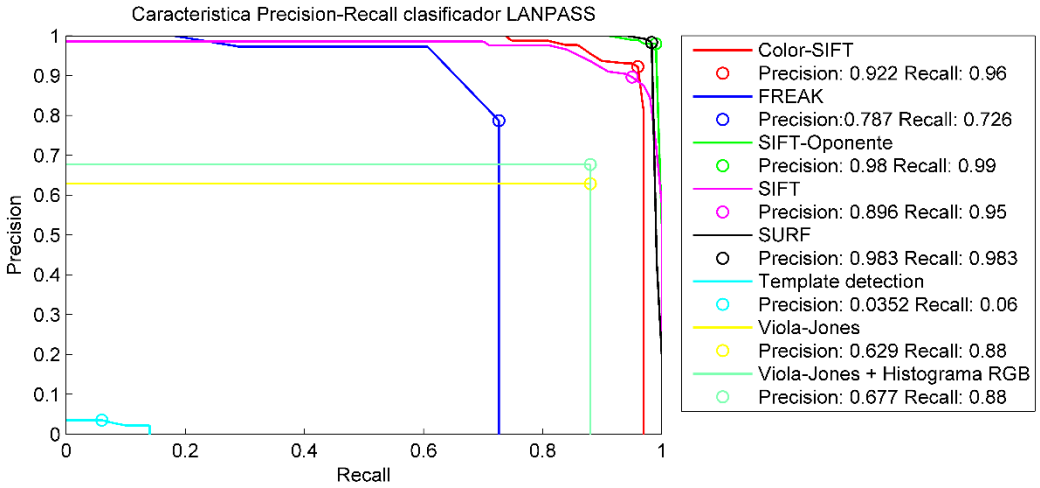

Caracteristica Precision-Recall clasificador ENTEL
Precision
Recall
Color-SIFT
Precision: 0.902 Recall: 0.435
FREAK
Precision:0.575 Recall: 0.6
SIFT-Oponente
Precision: 0.864 Recall: 0.447
SIFT
Precision: 0.913 Recall: 0.624
SURF
Precision: 0.908 Recall: 0.6
Template detection
Precision: 0.0278 Recall: 0.178
Viola-Jones
Precision: 0.776 Recall: 0.891
Viola-Jones + Histograma RGB
Precision: 0.849 Recall: 0.891

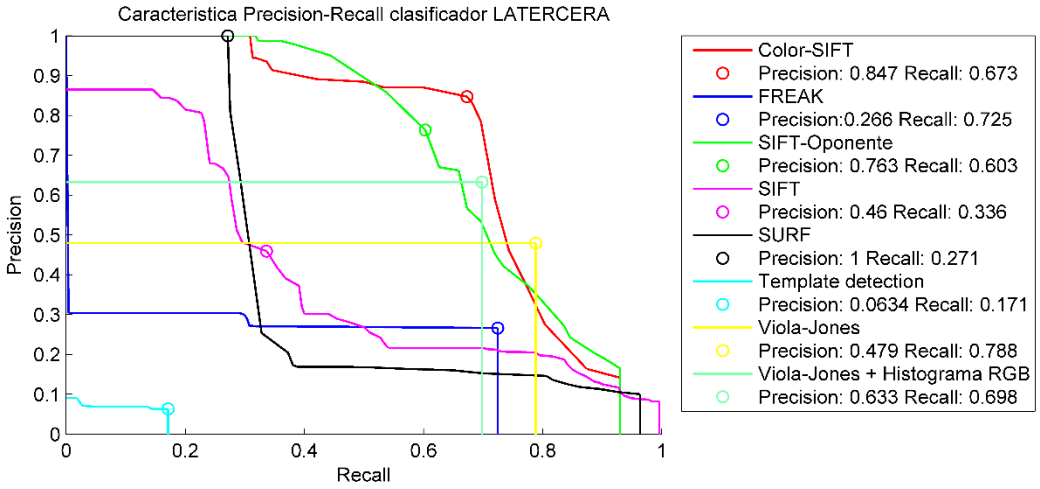

Caracteristica Precision-Recall clasificador FACEBOOK
Precision
Recall
Color-SIFT
Precision: 1 Recall: 0.0597
FREAK
Precision:0.714 Recall: 0.0156
SIFT-Oponente
Precision: 0.746 Recall: 0.175
SIFT
Precision: 0.386 Recall: 0.233
SURF
Precision: 0.233 Recall: 0.148
Template detection
Precision: 0.034 Recall: 0.0198
Viola-Jones
Precision: 0.827 Recall: 0.867
Viola-Jones + Histograma RGB
Precision: 0.907 Recall: 0.761

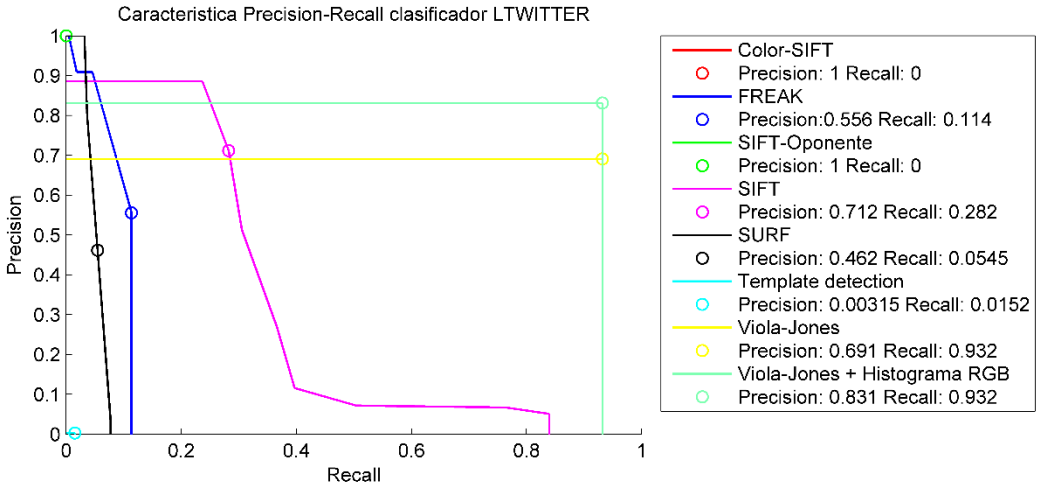

Caracteristica Precision-Recall clasificador FALABELLAL
Precision
Recall
Color-SIFT
Precision: 1 Recall: 0.333
FREAK
Precision:0.905 Recall: 0.564
SIFT-Oponente
Precision: 0.926 Recall: 0.889
SIFT
Precision: 0.369 Recall: 0.376
SURF
Precision: 0.821 Recall: 0.911
Template detection
Precision: 0.088 Recall: 0.099
Viola-Jones
Precision: 0.217 Recall: 0.149
Viola-Jones + Histograma RGB
Precision: 0.306 Recall: 0.149

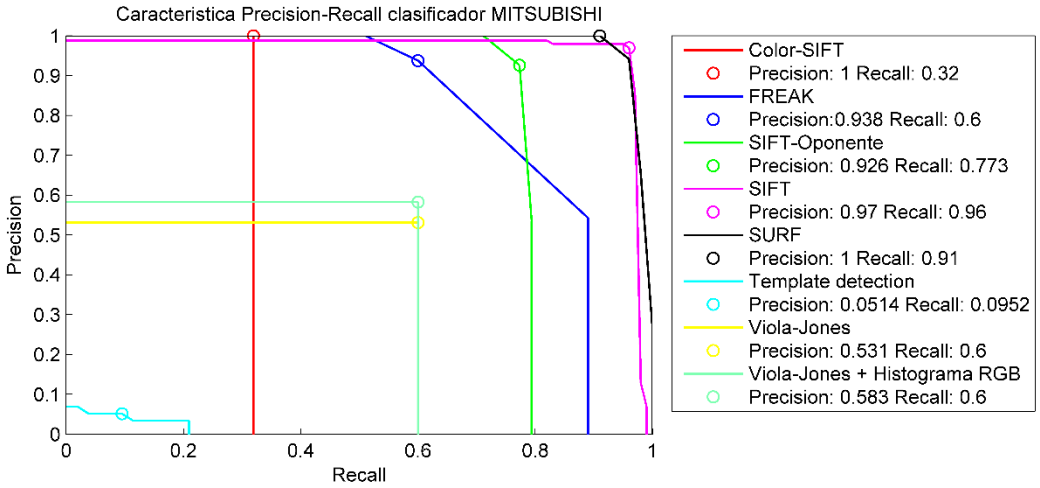

Caracteristica Precision-Recall clasificador FIAT
Precision
Recall
Color-SIFT
Precision: 1 Recall: 0.388
FREAK
Precision:0.846 Recall: 0.545
SIFT-Oponente
Precision: 0.967 Recall: 0.296
SIFT
Precision: 0.847 Recall: 0.822
SURF
Precision: 1 Recall: 0.505
Template detection
Precision: 0.034 Recall: 0.0198
Viola-Jones
Precision: 0.0704 Recall: 0.0495
Viola-Jones + Histograma RGB
Precision: 0.161 Recall: 0.0495

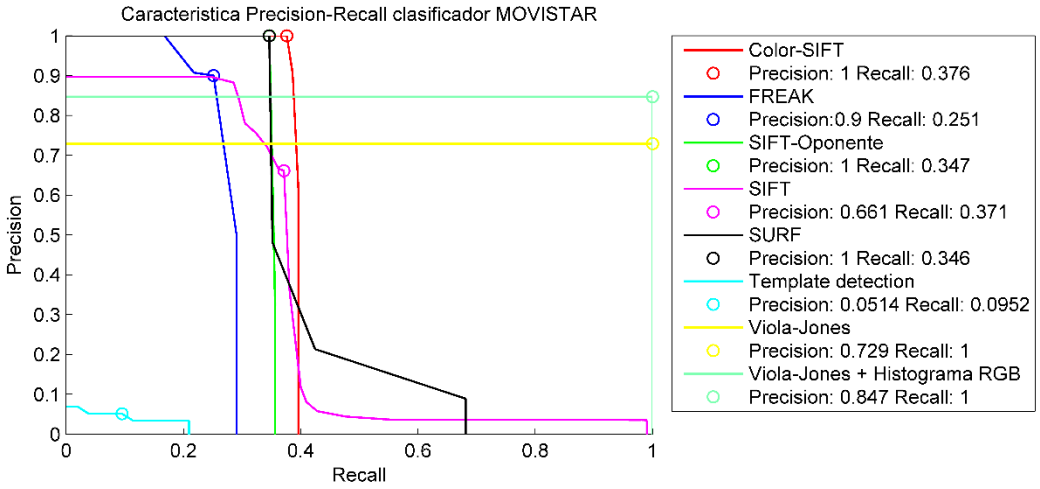

Caracteristica Precision-Recall clasificador JUMBO
Precision
Recall
Color-SIFT
Precision: 0.988 Recall: 0.856
FREAK
Precision:0.63 Recall: 0.463
SIFT-Oponente
Precision: 1 Recall: 0.825
SIFT
Precision: 0.987 Recall: 0.724
SURF
Precision: 1 Recall: 0.414
Template detection
Precision: 0.0161 Recall: 0.0935
Viola-Jones
Precision: 0.55 Recall: 0.206
Viola-Jones + Histograma RGB
Precision: 0.55 Recall: 0.206

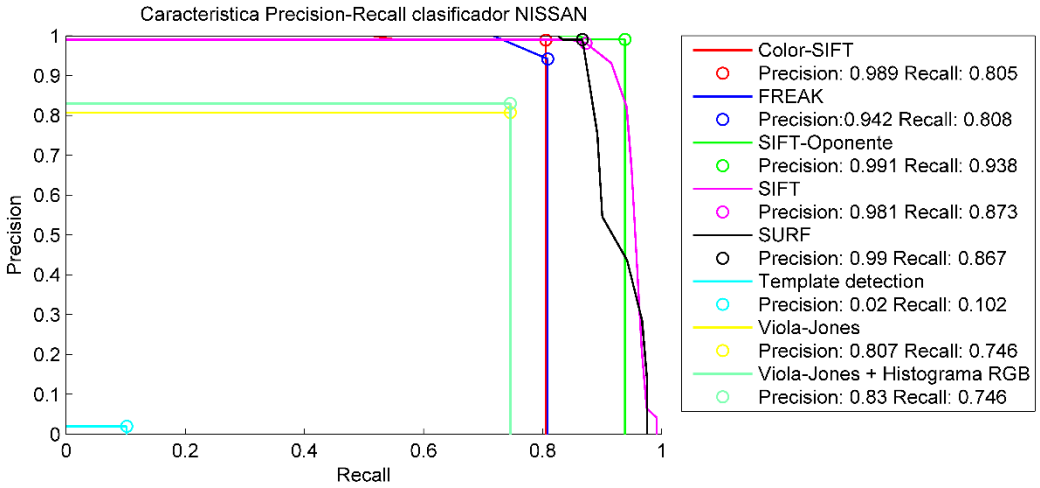

Caracteristica Precision-Recall clasificador LANPASS
Precision
Recall
Color-SIFT
Precision: 0.922 Recall: 0.96
FREAK
Precision:0.787 Recall: 0.726
SIFT-Oponente
Precision: 0.98 Recall: 0.99
SIFT
Precision: 0.896 Recall: 0.95
SURF
Precision: 0.983 Recall: 0.983
Template detection
Precision: 0.0352 Recall: 0.06
Viola-Jones
Precision: 0.629 Recall: 0.88
Viola-Jones + Histograma RGB
Precision: 0.677 Recall: 0.88

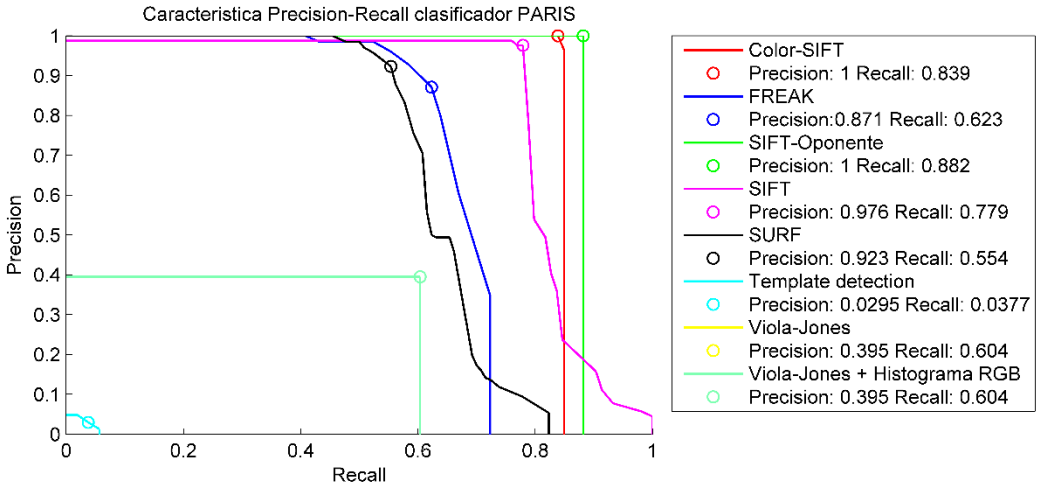

Caracteristica Precision-Recall clasificador LATERCERA
Precision
Recall
Color-SIFT
Precision: 0.847 Recall: 0.673
FREAK
Precision:0.266 Recall: 0.725
SIFT-Oponente
Precision: 0.763 Recall: 0.603
SIFT
Precision: 0.46 Recall: 0.336
SURF
Precision: 1 Recall: 0.271
Template detection
Precision: 0.0634 Recall: 0.171
Viola-Jones
Precision: 0.479 Recall: 0.788
Viola-Jones + Histograma RGB
Precision: 0.633 Recall: 0.698

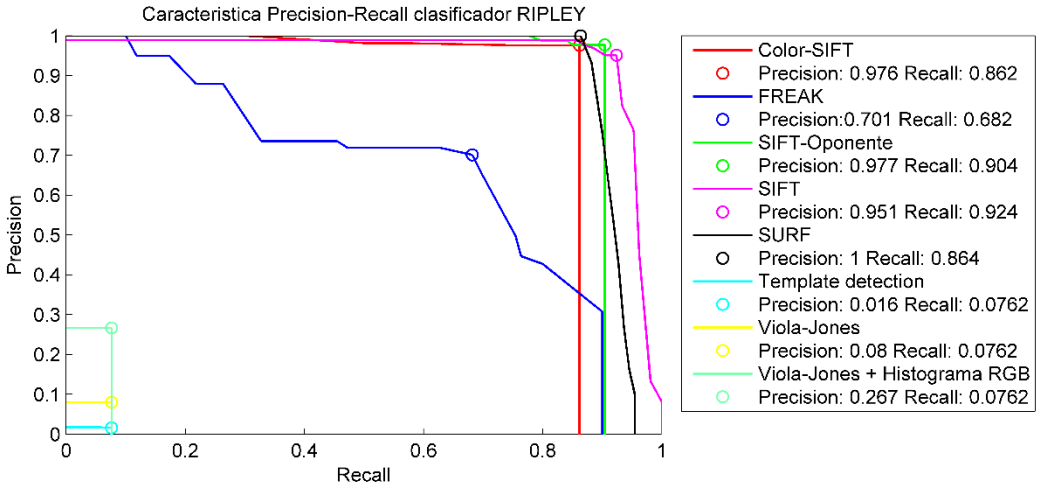

Caracteristica Precision-Recall clasificador LTWITTER
Precision
Recall
Color-SIFT
Precision: 1 Recall: 0
FREAK
Precision:0.556 Recall: 0.114
SIFT-Oponente
Precision: 1 Recall: 0
SIFT
Precision: 0.712 Recall: 0.282
SURF
Precision: 0.462 Recall: 0.0545
Template detection
Precision: 0.00315 Recall: 0.0152
Viola-Jones
Precision: 0.691 Recall: 0.932
Viola-Jones + Histograma RGB
Precision: 0.831 Recall: 0.932

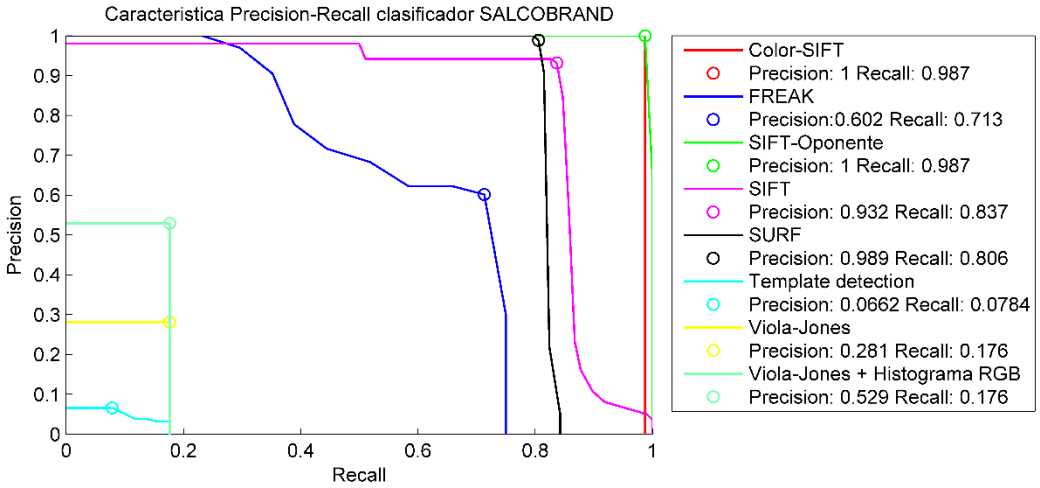

Caracteristica Precision-Recall clasificador MITSUBISHI
Precision
Recall
Color-SIFT
Precision: 1 Recall: 0.32
FREAK
Precision:0.938 Recall: 0.6
SIFT-Oponente
Precision: 0.926 Recall: 0.773
SIFT
Precision: 0.97 Recall: 0.96
SURF
Precision: 1 Recall: 0.91
Template detection
Precision: 0.0514 Recall: 0.0952
Viola-Jones
Precision: 0.531 Recall: 0.6
Viola-Jones + Histograma RGB
Precision: 0.583 Recall: 0.6

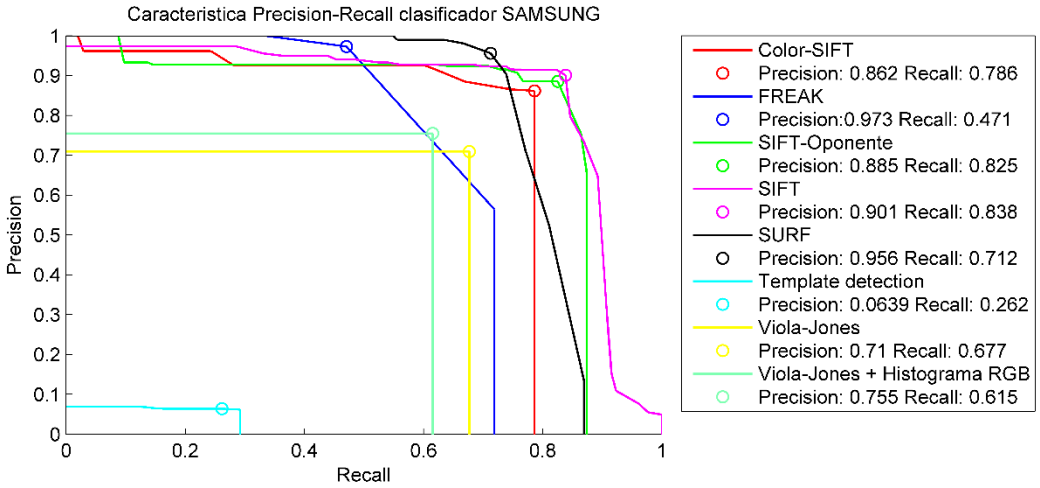

Caracteristica Precision-Recall clasificador MOVISTAR
Precision
Recall
Color-SIFT
Precision: 1 Recall: 0.376
FREAK
Precision:0.9 Recall: 0.251
SIFT-Oponente
Precision: 1 Recall: 0.347
SIFT
Precision: 0.661 Recall: 0.371
SURF
Precision: 1 Recall: 0.346
Template detection
Precision: 0.0514 Recall: 0.0952
Viola-Jones
Precision: 0.729 Recall: 1
Viola-Jones + Histograma RGB
Precision: 0.847 Recall: 1

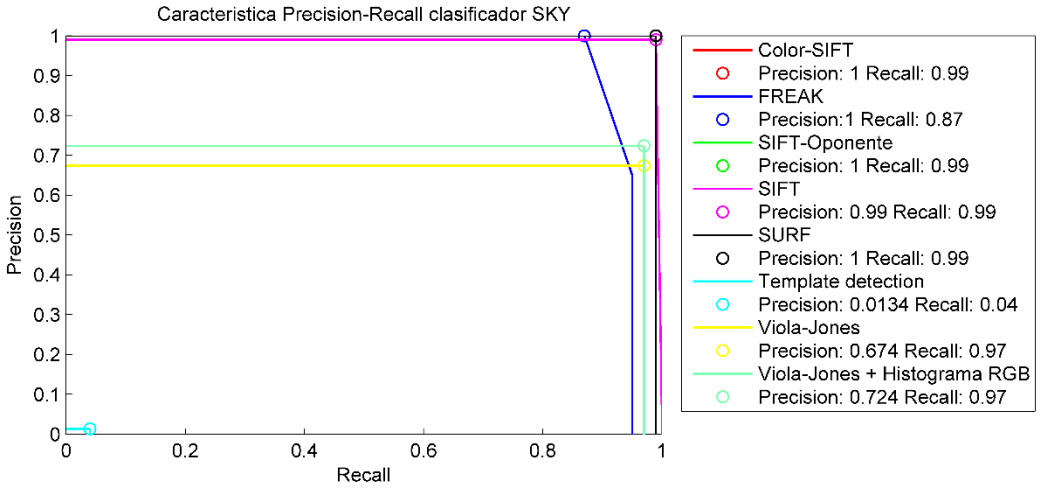

Caracteristica Precision-Recall clasificador NISSAN
Precision
Recall
Color-SIFT
Precision: 0.989 Recall: 0.805
FREAK
Precision:0.942 Recall: 0.808
SIFT-Oponente
Precision: 0.991 Recall: 0.938
SIFT
Precision: 0.981 Recall: 0.873
SURF
Precision: 0.99 Recall: 0.867
Template detection
Precision: 0.02 Recall: 0.102
Viola-Jones
Precision: 0.807 Recall: 0.746
Viola-Jones + Histograma RGB
Precision: 0.83 Recall: 0.746

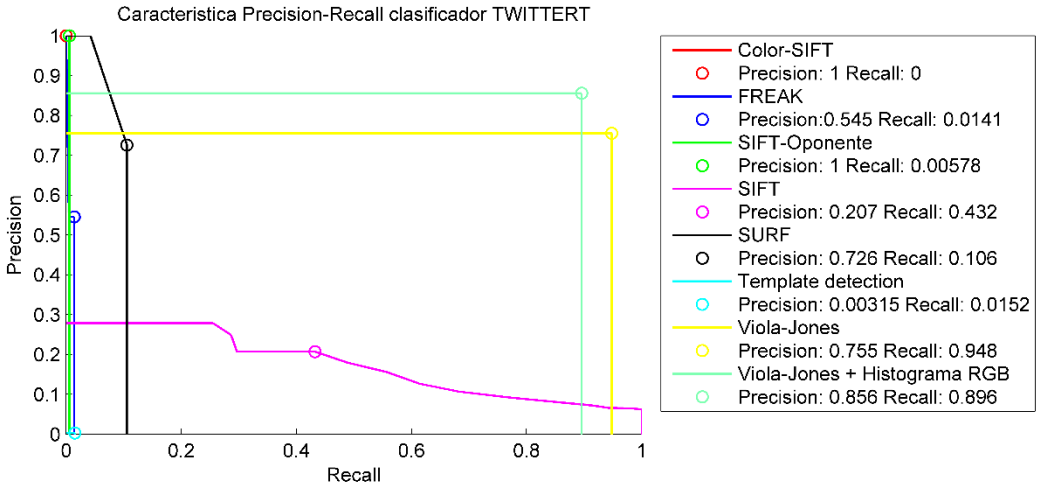

Caracteristica Precision-Recall clasificador PARIS
Precision
Recall
Color-SIFT
Precision: 1 Recall: 0.839
FREAK
Precision:0.871 Recall: 0.623
SIFT-Oponente
Precision: 1 Recall: 0.882
SIFT
Precision: 0.976 Recall: 0.779
SURF
Precision: 0.923 Recall: 0.554
Template detection
Precision: 0.0295 Recall: 0.0377
Viola-Jones
Precision: 0.395 Recall: 0.604
Viola-Jones + Histograma RGB
Precision: 0.395 Recall: 0.604

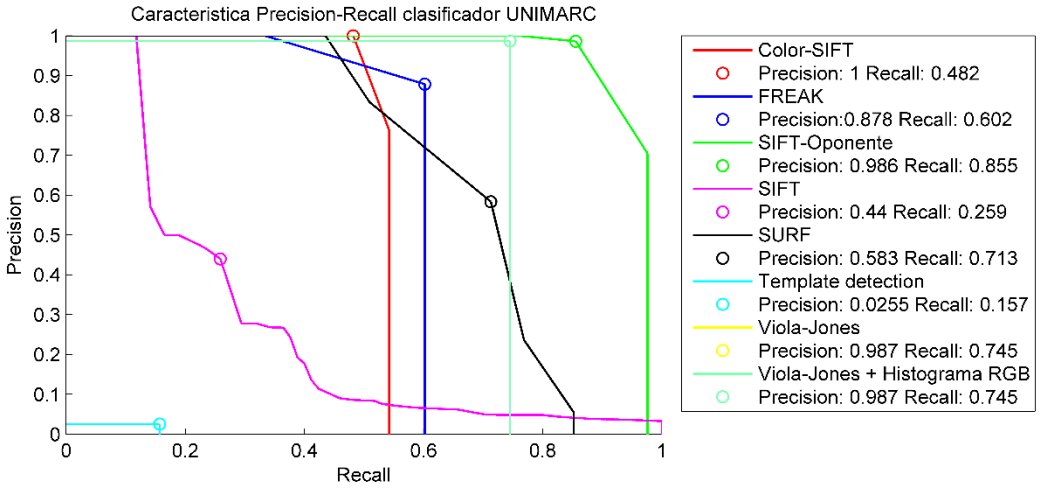

Caracteristica Precision-Recall clasificador RIPLEY
Precision
Recall
Color-SIFT
Precision: 0.976 Recall: 0.862
FREAK
Precision:0.701 Recall: 0.682
SIFT-Oponente
Precision: 0.977 Recall: 0.904
SIFT
Precision: 0.951 Recall: 0.924
SURF
Precision: 1 Recall: 0.864
Template detection
Precision: 0.016 Recall: 0.0762
Viola-Jones
Precision: 0.08 Recall: 0.0762
Viola-Jones + Histograma RGB
Precision: 0.267 Recall: 0.0762


Caracteristica Precision-Recall clasificador SALCOBRAND
Precision
Recall
Color-SIFT
Precision: 1 Recall: 0.987
FREAK
Precision:0.602 Recall: 0.713
SIFT-Oponente
Precision: 1 Recall: 0.987
SIFT
Precision: 0.932 Recall: 0.837
SURF
Precision: 0.989 Recall: 0.806
Template detection
Precision: 0.0662 Recall: 0.0784
Viola-Jones
Precision: 0.281 Recall: 0.176
Viola-Jones + Histograma RGB
Precision: 0.529 Recall: 0.176


Caracteristica Precision-Recall clasificador SAMSUNG
Precision
Recall
Color-SIFT
Precision: 0.862 Recall: 0.786
FREAK
Precision:0.973 Recall: 0.471
SIFT-Oponente
Precision: 0.885 Recall: 0.825
SIFT
Precision: 0.901 Recall: 0.838
SURF
Precision: 0.956 Recall: 0.712
Template detection
Precision: 0.0639 Recall: 0.262
Viola-Jones
Precision: 0.71 Recall: 0.677
Viola-Jones + Histograma RGB
Precision: 0.755 Recall: 0.615


Caracteristica Precision-Recall clasificador SKY
Precision
Recall
Color-SIFT
Precision: 1 Recall: 0.99
FREAK
Precision:1 Recall: 0.87
SIFT-Oponente
Precision: 1 Recall: 0.99
SIFT
Precision: 0.99 Recall: 0.99
SURF
Precision: 1 Recall: 0.99
Template detection
Precision: 0.0134 Recall: 0.04
Viola-Jones
Precision: 0.674 Recall: 0.97
Viola-Jones + Histograma RGB
Precision: 0.724 Recall: 0.97


Caracteristica Precision-Recall clasificador TWITTERT
Precision
Recall
Color-SIFT
Precision: 1 Recall: 0
FREAK
Precision:0.545 Recall: 0.0141
SIFT-Oponente
Precision: 1 Recall: 0.00578
SIFT
Precision: 0.207 Recall: 0.432
SURF
Precision: 0.726 Recall: 0.106
Template detection
Precision: 0.00315 Recall: 0.0152
Viola-Jones
Precision: 0.755 Recall: 0.948
Viola-Jones + Histograma RGB
Precision: 0.856 Recall: 0.896


Caracteristica Precision-Recall clasificador UNIMARC
Precision
Recall
Color-SIFT
Precision: 1 Recall: 0.482
FREAK
Precision:0.878 Recall: 0.602
SIFT-Oponente
Precision: 0.986 Recall: 0.855
SIFT
Precision: 0.44 Recall: 0.259
SURF
Precision: 0.583 Recall: 0.713
Template detection
Precision: 0.0255 Recall: 0.157
Viola-Jones
Precision: 0.987 Recall: 0.745
Viola-Jones + Histograma RGB
Precision: 0.987 Recall: 0.745