UNIVERSIDAD NACIONAL DE CÓRDOBA
Facultad de Ciencias Exactas Físicas y Naturales
Doctorado en Ciencias de la Ingeniería

Tesis Doctoral

Aplicación en agricultura de precisión de esquemas actuales de reconocimiento visual

Autor: Ing. Javier A. REDOLFI
Director: Prof. Dr. Julián PUCHETA

Mayo de 2018

Aplicación en agricultura de precisión de esquemas actuales de reconocimiento visual

por

Ing. Javier A. REDOLFI

Prof. Dr. Julián PUCHETA

Director

COMISIÓN ASESORA:

Prof. Dr. Julián PUCHETA

FCEFyN - UNC

Prof. Dr. Víctor SAUCHELLI

FCEFyN - UNC

Prof. Dr. Jorge SÁNCHEZ

FaMAF- UNC

Esta Tesis fue enviada a la Facultad de Ciencias Exactas Físicas y Naturales de la Universidad Nacional de Córdoba para cumplimentar los requerimientos de obtención del grado académico de Doctor en Ciencias de la Ingeniería.

Córdoba, Argentina
Mayo de 2018

UNIVERSIDAD NACIONAL DE CORDOBA
Facultad de Cs. Exactas, Fisicas y Naturales

ACTA DE EXAMENES

Libro: 00001 Acta: 04287 Hoja 01/01
LLAMADO: 1 11/05/2018
CATEDRA - MESA:

DI002 TESIS DOCTORADO EN CIENCIAS DE LA INGENIERIA

NUMERO	APELLIDO Y NOMBRE	DOCUMENTO	INGRESO	COND.	NOTA	FIRMA
31385983	REDOLFI, Javier Andrés	DNI: 31385983	2013	T	Aprobado	

WOLFMANN, Gustavo - DI PERSIA, Leandro - ALONSO I ALEMANY, Laura - ARAGUÁS, GASTÓN - MALDONAL

Observaciones:

Córdoba, ____/____/_____.
Certifico que la/s firma/s que ha/n sido puesta/s en la presente Acta pertenece/n a:

Inscriptos	Ausentes	Examinados	Reprobados	Aprobados
1	0	1	0	1

25/04/2018 10:07:26

Libro/Acta: 0000104287 Hoja: 01/ 01

Dedicado a Olivia, Victoria y a mi familia

Agradecimientos

Esta tesis no hubiera posible sin el incentivo y apoyo que me brindó mi familia desde el inicio de mi carrera, por eso mi primer agradecimiento es para ellos. También para Victoria por su paciencia en estos largos años.

A todos mis compañeros del Centro de Investigación en Informática para la Ingeniería, a los que siguen y a los que buscaron otros rumbos los cuales me brindaron un cálido y humano lugar de trabajo. Directores, investigadores y becarios con los cuales compartimos momentos lindos y otros no tanto, proyectos, viajes, ideas y discusiones.

También quiero agradecer a mi director Julián Pucheta por todos los consejos, ayuda, discusiones y apoyo en todo este proceso.

Y por último quiero agradecer a Jorge Sánchez, persona que gracias a su dedicación y consejos hizo posible que pueda llegar a esto.

Resumen

La agricultura es una actividad de gran importancia estratégica, base fundamental para el desarrollo autosuficiente y la riqueza de las naciones. Según datos de la Organización de las Naciones Unidas para la Alimentación y la Agricultura entre el 20 y el 40 % de la producción agrícola anual se pierde debido a pestes y enfermedades, a pesar de la utilización de 2 millones de toneladas de pesticidas. Por lo tanto resulta necesario tener una nueva mirada sobre la agricultura y las técnicas utilizadas, en busca de lograr un aumento en la producción para satisfacer a una demanda creciente, pero siendo lo más amigables posible con el ambiente. Esta nueva mirada debe acercar disciplinas como la robótica, la visión por computadora y el aprendizaje de máquina a las técnicas agropecuarias.

La agricultura de precisión es un modelo que usa tecnologías como análisis satelital y datos de sistemas de posicionamiento global, en conjunto con sensores ubicados en los campos y fotografías tomadas por drones o robots para obtener información sobre el estado del cultivo. Con esta información el productor puede tener una idea gráfica y un enfoque más detallado y oportuno para la toma de decisiones. Pero en la actualidad, debido al rápido avance de la tecnología, la variedad de sensores disponibles es cada vez mayor y también su resolución, lo que se traduce en un aumento en la cantidad de datos a procesar y en la diversidad en la naturaleza de los mismos. Por lo tanto para poder hacer frente a esto, resulta necesario el estudio de los modelos más recientes propuestos en la literatura y el desarrollo de nuevos modelos para lograr un correcto aprovechamiento de estas nuevas y abundantes formas de información.

En esta tesis se plantea la aplicación de modelos de clasificación y recuperación de imágenes recientemente propuestos en la literatura como son vectores de Fisher y redes neuronales convolucionales a problemas de agricultura de precisión, como la clasificación de especies de plantas y de variedades de semillas de una misma especie utilizando imágenes RGB y la clasificación de uso de suelo a través de imágenes obtenidas con radares de apertura sintéticos. Con la inclusión de tales modelos se logra otorgar un mayor nivel de robustez y escalabilidad a los sistemas lo cual se traduce en un aumento en la exactitud de la solución de estos problemas. Para la validación de la propuesta se realizaron experimentos sobre los problemas mencionados y se obtuvieron soluciones que compiten con el estado del arte y en algunos casos las superan. Además se desarrollan nuevos modelos en base a los existentes, específicamente se crea un nuevo modelo llamado vectores de Fisher de la familia exponencial que extiende la codificación vectores de Fisher. Esto último, además de mejorar la exactitud, permite la aplicación de estos modelos en nuevos problemas.

Por último, después del desarrollo de la tesis quedan temas de investigación abiertos tanto desde el punto de vista teórico, como por ejemplo la extensión a dominios de entrada más amplios de los métodos desarrollados, y también en el ámbito de nuevas aplicaciones, como puede ser ayudar a resolver problemas urgentes de la región relacionados con la conservación del medio ambiente.

Palabras Clave: Agricultura de Precisión, Vectores de Fisher, Redes Neuronales Convolucionales, Clasificación de Imágenes

Resumo

A agricultura é uma atividade de grande importância estratégica, uma base fundamental para o desenvolvimento auto-suficiente e a riqueza das nações. De acordo com dados da Organização das Nações Unidas para a Alimentação e Agricultura, entre 20 e 40 % da produção agrícola anual é perdida devido a pragas e doenças, apesar do uso de 2 milhões de toneladas de pesticidas. Por conseguinte, é necessário dar uma nova olhada na agricultura e as técnicas utilizadas para alcançar um aumento da produção para atender a demanda crescente, sendo o mais ecológico possível. Essa nova perspectiva deve trazer disciplinas como a robótica, a visão por computador e a aprendizagem de máquinas para técnicas agrícolas.

A agricultura de precisão é um modelo que usa tecnologias como análise satélital e dados de GPS, em conjunto com sensores localizados em campos e fotografias de drones para obter informações sobre o estado dos cultivos. Com esta informação, o produtor pode ter uma idéia gráfica e uma abordagem mais detalhada e oportuno para a toma de decisões. Atualmente, devido ao rápido avanço da tecnologia, a variedade de sensores disponíveis e sua resolução, se traduz em um aumento na quantidade de dados a serem processados e na natureza dos mesmos. Portanto, para lidar com isso, é necessário estudar os modelos mais recentes propostos na literatura e o desenvolvimento de novos modelos para alcançar o correto uso dessas novas e abundantes formas de informação.

Esta tese apresenta a aplicação de modelos de classificação e recuperação de imagens recentemente propostos na literatura, como FV e CNN sobre problemas de agricultura de precisão, como a classificação de espécies de plantas e variedades de sementes da mesma espécie usando imagens RGB e a classificação de uso da terra através de imagens obtidas com radares de abertura sintética. Com a inclusão de tais modelos, é possível dar um maior nível de robustez e escalabilidade aos sistemas, o que resulta em um aumento na precisão da solução desses problemas. Para a validação da proposta, foram realizados experimentos sobre os problemas mencionados e foram obtidas soluções que competem com o estado da arte e, em alguns casos, as excedem. Além disso, novos modelos são desenvolvidos com base em modelos existentes, criando especificamente um novo modelo chamado eFV que amplia a codificação FV. O último, além de melhorar a precisão, permite a aplicação desses modelos em novos problemas.

Finalmente, após o desenvolvimento da tese, os tópicos de pesquisa permanecem abertos tanto do ponto de vista teórico, como por exemplo a extensao a domínios de entrada mais amplos dos métodos desenvueltos e também no campo de novas aplicações, que podem ajudar a resolver problemas ambientais urgentes da região relacionados com a com a conservação do meio ambiente.

Palavras-chave: Agricultura de Precisão, Vetores de Fisher, Redes Neurais Convolucionais, Classificação de Imagens

Abstract

Agriculture is an activity of great strategic importance, a fundamental basis for the self-sufficient development and wealth of nations. According to data from the Food and Agriculture Organization of the United Nations, between 20 and 40 % of annual agricultural production is lost due to pests and diseases, despite the use of 2 million tons of pesticides. It is therefore necessary to take a new look at agriculture and the techniques used to achieve an increase in production to meet growing demand, and be as environmentally friendly as possible. This new perspective should bring disciplines such as robotics, computer vision and machine learning for agricultural techniques.

Precision Agriculture is a model that uses technologies such as satellite analysis and GPS data, along with sensors located in fields and drone photographs to obtain information on the state of crops. With this information, the producer can have a graphic idea and a more detailed and timely approach to decision making. Today, due to the rapid advancement of technology, the variety of sensors available and their resolution, translates into an increase in the amount of data to be processed and the nature of the data. Therefore, to deal with this, it is necessary to study the most recent models proposed in the literature and the development of new models to achieve the correct use of these new and abundant forms of information.

This thesis proposes the application of recently proposed image classification and image retrieval models in the literature, such as FV and CNN on precision agriculture problems, such as the classification of plant species and seed varieties of the same species using RGB images and the classification of land use through images obtained with synthetic aperture radars. With the inclusion of such models, it is possible to give a greater level of robustness and scalability to the systems, which results in an increase in the precision of the solution of these problems. For the validation of the proposal, experiments were carried out on the mentioned problems and solutions were obtained that compete with the state of the art and, in some cases, exceed them. In addition, new models are developed based on existing models, specifically a new model called eFV is created that extends the FV coding. The latter, in addition to improving accuracy, allows the application of these models in new problems.

Finally, after the development of the thesis, the research topics remain open both from a theoretical point of view, for example the extension to wider domains of the developed methods and also in the field of new applications, which can help solve environmental problems of the region like the conservation of the environment.

Keywords: Precision Agriculture, Fisher Vectors, Convolutional Neural Networks, Image Classification

Índice general

Lista de acrónimos

En esta tesis se decidió que la mayoría de los acrónimos estén definidos por sus siglas en inglés para que sea directa su correlación con los términos usados en la literatura.

AP Agricultura de Precisión

BinSIFT Transformación de Características Invariantes ante Escala Binarizadas

BoVW Bolsa de Palabras Visuales

BRIEF Características Elementales Independientes Robustas Binarias

CNN Redes Neuronales Convolucionales

CV Visión por Computadora

CWC Clasificador Complejo de Wishart

DCOV Descriptores de Covarianza

DL Aprendizaje Profundo

eFV Vectores de Fisher de la Familia Exponencial

EM Esperanza-Maximización

FIM Matriz de Información de Fisher

GPU Unidad de Procesamiento Gráfico

FK Núcleo de Fisher

FV Vectores de Fisher

GMM Modelo Mezcla de Gaussianas

INTA Instituto Nacional de Tecnología Agropecuaria

LBP Patrones Locales Binarios

LBPH Histogramas de Patrones Locales Binarios

ML Aprendizaje de Máquina

PCA Análisis de Componentes Principales

PSD Positiva Semi-Definida

pdf Función de Distribución de Probabilidad

PolSAR Radar de Apertura Sintético Polarizado

SIFT Transformación de Características Invariantes ante Escala

SPD Simétricas Positiva Definida

SV Super Vector

SVM Máquinas de Soporte Vectorial

VLAD Vector de Descriptores Localmente Agregados

Capítulo 1

Introducción

La *agricultura* del latín agri "campo" y cultura "cultivo" o "crianza" es el conjunto de técnicas y conocimientos para cultivar la tierra y la parte del sector primario que se dedica a ello [1]. En ella se engloban los diferentes trabajos de tratamiento del suelo y los cultivos de vegetales. Esta comprende todo un conjunto de acciones humanas que transforma el medio ambiente natural y las actividades que abarca dicho sector tienen su fundamento en la explotación de los recursos que la tierra origina, favorecida por la acción del ser humano: alimentos vegetales como cereales, frutas, hortalizas, pastos cultivados y forrajes; fibras utilizadas por la industria textil; cultivos energéticos etc.

Es una actividad de gran importancia estratégica, base fundamental para el desarrollo autosuficiente y riqueza de las naciones. En la Argentina, la agricultura representa el 11.4 % del Producto Bruto Interno (PBI) [2]. En línea con el crecimiento de la población mundial la demanda de producción de alimentos está en aumento [RMWF13] pero según datos de la Organización de las Naciones Unidas para la Alimentación y la Agricultura entre el 20 y el 40 % de la producción agrícola anual se pierde debido a pestes y enfermedades, a pesar de la utilización de 2 millones de toneladas de pesticidas [K$^+$17].

Por estas razones, resulta necesario tener una nueva mirada sobre la agricultura y las técnicas utilizadas, para lograr producir un aumento en la producción de alimentos para satisfacer a una demanda creciente, pero tratando de ser lo más amigables posible con el medio ambiente. En esta nueva mirada debe haber un acercamiento entre las técnicas agropecuarias y las nuevas disciplinas tecnológicas como son la robótica, la Visión por Computadora (CV, por su denominación en inglés) y el Aprendizaje de Máquina (ML, por su denominación en inglés).

1.1. Agricultura de precisión

La Agricultura de Precisión (AP) es una clase de técnica agrícola que integra tecnologías como sistemas de información geográfica y sistemas de posicionamiento global, en conjunto con sensores remotos ubicados estratégicamente en los campos y registros de imágenes tomadas por satélites o drones para obtener información sobre el estado del cultivo. A diferencia del manejo tradicional de campo, que consiste en aplicar el mismo tratamiento a todo el campo, con la información obtenida con los diversos sensores la AP permite manejar la variabilidad dentro de un campo aplicando un tratamiento selectivo dependiendo del estado del cultivo en cada sector [BBH$^+$98].

El concepto fundamental de la AP es el manejo específico de sitio, que consiste en realizar los procesos agrícolas cómo por ejemplo fertilización o fumigación con una tasa variable de apli-

[1] Definición de la real academia española.

[2] `https://www.cia.gov/library/publications/the-world-factbook/fields/2012.html#ar`

cación dependiente de las necesidades de cada sector del campo. Este manejo permite reducir la cantidad de producto aplicado, pero optimizando los rendimientos. Esto no solo es beneficioso por la reducción del uso de agroquímicos que baja los costos del productor, sino también para el medio ambiente. Para realizar este manejo focalizado es necesario que las máquinas estén dotadas con sensores capaces de realizar las mediciones, con actuadores capaces de realizar una aplicación variable y con algoritmos capaces de generar la información para conectar a ambos.

Esta técnica surgió en la década del 80 y en la Argentina el Instituto Nacional de Tecnología Agropecuaria (INTA) empezó con las primeras pruebas hacia fines de los 90 [MAK13]. Y aunque la adopción de este tipo de tecnologías viene aumentando en forma creciente a lo largo de los años, hay una pequeña cantidad de sistemas que aplican los productores en las tareas rurales como son los monitores de rendimiento, monitores de siembra, dosificadores variables, banderilleros satelitales, entre otros [SVV15]. Pero se espera, que en un futuro, el rango de aplicaciones se incremente en forma exponencial de la mano de las nuevas tecnologías desarrolladas en el ámbito de la electrónica, el software, las comunicaciones, la conectividad y la robotización a partir del desarrollo y aplicación de sensores en conjunto con algoritmos capaces de identificar objetos, plantas, estado de humedad y nutrición del suelo; también medir variables climáticas como humedad relativa, temperatura, velocidad del viento, lluvia, evapotranspiración actual y potencial, o la capacidad para escanear granos y detectar daño mecánico, impurezas, contenido de aceite y proteína en la misma máquina cosechadora; o nuevos sensores que guíen a las cosechadoras para recoger solamente la fruta madura, o sensores de insectos para grano almacenados; y también satélites o nanosatélites de alta resolución espacial y temporal para generar información a gran escala del estado de diferentes variables en los campos. [Esp16, Bau16, RR16]

Aunque la aceptación y el crecimiento de la AP en algunas aplicaciones como las nombradas más arriba ha sido rápido, se necesita que ciertos requerimientos se cumplan para ayudar al desarrollo completo e implementación de esta técnica en todos los ámbitos agrícolas. Dentro de ellos puede nombrar a la investigación y desarrollo continuo de algoritmos, el acceso a datos y sistemas de bajo costo y por último una siguiente fase de entrenamiento y transferencia de tecnología para acelerar la aceptación e implementación de esta tecnología en el sector agrícola. [BBH$^+$98]

1.2. Visión por computadora

La CV es el área del conocimiento que estudia cómo las computadoras pueden extraer conceptos de alto nivel desde imágenes o videos digitales y su objetivo principal es crear herramientas que resuelvan tareas que el sistema visual humano resuelve en forma satisfactoria. Las tareas que incluye son métodos para adquirir, procesar, analizar y entender imágenes digitales. El entendimiento, en este contexto, significa transformar las imágenes visuales en descripciones del mundo que puedan ser usadas en otros procesos para poder realizar la acción apropiada. Este entendimiento puede ser visto como el desentramado de la información simbólica desde los datos que forman las imágenes usando modelos construidos con ayuda de la geometría, la física, la estadística y la teoría del aprendizaje.

Como una disciplina científica, la CV está interesada en la teoría detrás de los sistemas artificiales que extraen información de las imágenes. Los datos de las imágenes pueden tomar muchas formas, como secuencias de video, vistas de múltiples cámaras, datos multidimensionales de un escáner médico o datos obtenidos con cámaras multiespectrales o hiperespectrales. Como una disciplina tecnológica, la CV busca aplicar estas teorías y modelos para las construcción de sistemas de visión por computadora.

1.3. Aplicaciones de la visión por computadora en agricultura de precisión

Las tecnologías agropecuarias basadas en CV no solo son el futuro, también son un desarrollo presente, el cual permite mejorar nuestras vidas poniendo en nuestra mesa comida más sana, producida con menos recursos y en mayores cantidades, en orden de adaptarse al abrumador crecimiento de la población mundial [K$^+$17]. El análisis gráfico tiene un rol muy importante en la creación de planes apropiados para el manejo agrícola. Por ejemplo, la humedad de suelo puede ser detectada desde un análisis utilizando imágenes, una vez realizado el estudio es posible direccionar máquinas de riego a las áreas en donde la humedad de suelo es baja. También con el uso de imágenes aéreas se puede detectar las clases de pestes en los cultivos o también algún defecto en el cultivo para aplicar un tratamiento oportuno en las secciones problemáticas en vez de tener una vista en conjunto de la plantación que omita tales detalles. Otra aplicación es el monitoreo del crecimiento, rendimiento y salud del cultivo, con él se puede obtener una gran cantidad de datos los cuales pueden ser explotados para optimizar el riego, reducir las fumigaciones y planear una correcta fecha de cosecha.

En la actualidad tecnologías como la robótica y la CV están al alcance de la mano del productor y con ellas se está produciendo un nuevo salto en la calidad de las técnicas agropecuarias, en conjunto con un aumento en la producción pero con un menor desgaste de la naturaleza. Dentro de estas nuevas tecnologías se puede nombrar la facilidad de accesos a robots móviles o robots voladores [3], dotados de cámaras de muy alta resolución y en variados espectros de frecuencia. Estas máquinas voladoras pueden tomar numerosas fotos de bosques, árboles y cultivos, las cuales posteriormente son procesadas con algoritmos de CV y los resultados son entregados a los dueños y administradores para ayudarlos a tomar las mejores decisiones y reducir la probabilidad de una mala operación [TS15, K$^+$17]. Un ejemplo de esto es la empresa francesa Airinov [4] la cual se dedica a realizar el procesamiento de estas imágenes para generar mapas de estrés hídrico, índice verde, contenido de nitrógeno, entre otros.

Otra fuente de datos muy importante es la que se puede obtener usando sensores montados en satélites, como pueden ser cámaras hiperespectrales, infrarrojas o radares como los Radar de Apertura Sintético Polarizado (PolSAR, por su denominación en inglés). Este tipo de sensado se conoce como sensado remoto y con el procesamiento de estos datos se puede obtener información de las propiedades biofisiológicas de los cultivos [MIB97]. Las imágenes tomadas por los satélites pueden ser mejoradas para mostrar los tipos de suelo y que tipos de cultivos pueden soportar, permitiendo un avance en la distribución y variedad de los cultivos [Bau16]. El acceso a este tipo de imágenes se ha simplificado mucho, no solo por la puesta en disponibilidad al público para aplicaciones civiles de datos de satélites nacionales e internacionales, sino también por la aparición de nuevas empresas que brindan estos servicios, por ejemplo en Argentina, la empresa Satellogic [5] está desarrollando su propia constelación de satélites con diversas cámaras a bordo (RGB, hiper-espectrales y multi-espectrales). Otro desarrollo en el país es la misión SAOCOM [6], comandada por la CONAE, cuyo objetivo es la puesta en órbita de 4 de satélites con sensores SAR, de los cuales los dos primeros serán radares PolSAR. Se planea que el primer lanzamiento sea realizado en el año 2018 y entre las aplicaciones propuestas para estos radares figuran como primeras las actividades agrícolas y forestales, por ejemplo para la generación de mapas de cultivos y mapas forestales. Por lo dicho resulta de particular interés el estudio de las herramientas necesarias para aprovechar estos datos y la transferencia de estos conocimientos a los productores.

[3] https://www.sensefly.com/home.html

[4] https://www.airinov.fr/

[5] https://www.satellogic.com/

[6] http://www.conae.gov.ar/index.php/espanol/misiones-satelitales/saocom/objetivos

Otra práctica que está creciendo es el uso de fumigación selectiva, la cual consiste en dotar a la máquina fumigadora con un "sensor de verde" [7] para detectar la presencia de malezas y solo aplicar pesticidas en las zonas verdes. Con esto se produce una reducción en la cantidad de agroquímicos, en los costos y un menor impacto en el medio ambiente. En la actualidad estos tipos de sensores solo detectan la presencia de verde, pero con las nuevas tecnologías estos sistemas pueden ser mejorados con algoritmos que discriminen entre cultivos y malezas [HO14] para una aplicación selectiva no solo de pesticidas sino también de fertilizantes [TBP+08] o la realización de otras tareas. En línea con esto, existen desarrollos de robótica móvil aplicada en la agricultura como por ejemplo el robot BoniRob [RBD+09], el cual actualmente permite el fenotipado autónomo pero se planea en un futuro dotarlo de actuadores que permitan realizar diversas tareas en forma autónoma como la medición de la compactación de suelo o la estampación mecánica de malezas. Para lograr que estas ideas se conviertan en aplicaciones reales, es necesario el desarrollo de algoritmos capaces de diferenciar entre distintas especies de plantas, capaces de diferenciar formatos de hojas para contarlas o capaces de medir el tamaño de una planta y además con un bajo tiempo de procesamiento que permita controlar los actuadores en tiempo real.

También el uso de semillas certificadas juega un rol muy importante en el incremento de la calidad y cantidad del cultivo, por lo tanto antes de sembrar una variedad de semillas es muy importante confirmar la variedad a utilizar [PPAFS12]. El análisis de semillas en el proceso de producción de granos debe ser realizado en diferentes etapas del proceso, incluyendo la producción de semillas, la calificación del cereal para industrialización o comercialización y también durante las investigaciones científicas para mejoras de las especies. Otras veces este análisis es obligatorio, por ejemplo en Argentina, por reglamentación, antes de la comercialización de distintos tipos de granos se debe realizar el análisis de una pequeña muestra del lote a comercializar [GVC05]. Esta actividad es clave para contribuir al agregado de valor al cultivo [GNVC02] y también porque el uso final de los granos depende del tipo y variedad de semilla específico [PP13]. También este puede brindar conocimiento adicional sobre el proceso de producción, control de la calidad de las semillas y en la identificación de las impurezas [PKS+13].

Además, las técnicas desarrolladas para la AP tienen otros campos de aplicación, como por ejemplo en el cuidado del bosque nativo. En los últimos años, ha habido un creciente interés en el problema de clasificación de especies de plantas usando información visual [CPB15, KT15, GJB+14, SBG14]. Algunas razones de esto son el gran número de especies en peligro de extinción y las altas tasas de deforestación debidas al corrimiento de la frontera agropecuaria y a un pobre planeamiento urbano. Las plantas tienen un rol crucial para la vida en la tierra y su descuido puede causar problemas irreversibles a la sociedad, como el calentamiento global, pérdida de la biodiversidad y daño ambiental [CPB15, WSM+14]. Con las imágenes obtenidas por drones o también con el análisis de imágenes satelitales, se puede realizar un mapeo temporal de la evolución de los bosques nativos, para evitar el desmonte y elaborar un plan de acción para la recuperación de los mismos.

En la actualidad, debido al rápido avance de la tecnología, la variedad de sensores disponibles es cada vez mayor y también su resolución, lo que se traduce en un aumento en la cantidad de datos a procesar y en la diversidad en la naturaleza de los mismos debida a los diferentes tipos de información que estos entregan. Por lo tanto para poder hacer frente a esto, resulta necesario el estudio de los modelos más recientes propuestos en la literatura y el desarrollo de nuevos modelos para lograr un correcto aprovechamiento de estas nuevas y abundantes formas de información.

[7] https://agriculture.trimble.com/precision-ag/products/weedseeker/?lang=es

1.4. Objetivo

En esta tesis se plantea la aplicación de modelos de clasificación y recuperación de imágenes recientemente propuestos en la literatura, específicamente el de Vectores de Fisher (FV, por su denominación en inglés) y el de Redes Neuronales Convolucionales (CNN, por su denominación en inglés), a problemas de agricultura de precisión. Se cree que la incorporación de tales modelos otorgará un mayor nivel de robustez y escalabilidad a los sistemas lo cual se traducirá en un aumento en la exactitud de los mismos. Además se plantea el desarrollo de nuevos modelos en base a los existentes, en particular la extensión de FV; esto aparte de producir mejores resultados, permitirá la aplicación de estos modelos en nuevos problemas. Hasta el momento, la mayoría de las soluciones a estos problemas involucraban técnicas probabilísticas y de procesamiento de imágenes, además muchas veces requerían la intervención de un usuario, pero el uso de estos modelos basados en ML no ha sido muy explorado en problemas de agricultura de precisión o su aplicación es aún muy incipiente.

1.5. Contribuciones

A continuación se enumeran las contribuciones principales de esta tesis, las cuales son desarrolladas en profundidad en los capítulos siguientes.

La primera contribución es la generalización del formalismo de FV a una amplia familia de distribuciones conocida como *familia exponencial* (capítulo 3). Ya que los miembros de esta familia están definidos sobre una amplia variedad de dominios de entrada, este nuevo modelo, llamado Vectores de Fisher de la Familia Exponencial (eFV, por su denominación en inglés), provee un entorno unificado desde el cual pueden ser derivadas representaciones flexibles y poderosas. Además el código usado para computar estas representaciones se encuentra disponible para descarga en la página web del proyecto [8] para su uso por parte de la comunidad en general.

Luego utilizando la representación eFV se realiza un análisis experimental de su uso para la identificación de plantas (capítulo 5). Se compara la codificación de diferentes descriptores usando dicha representación y se evalúa la exactitud sobre conjuntos de datos públicos. También se comparan estos resultados con métodos de estado del arte presentados en la literatura demostrando que con la representación eFV se obtienen buenos resultados.

Otro problema abordado es el de identificación de variedades de semillas de trigo (capítulo 6). La solución propuesta, planteo fundamental de esta tesis, es el uso de técnicas actuales de clasificación de imágenes como son eFV y CNN. Con estas se logra una exactitud del 95 % en la clasificación de un conjunto de imágenes de semillas. Hasta lo que se conoce, el uso de estos modelos aún no había sido explorado en este tipo de problemas. Como contribución adicional se recolectó un conjunto de imágenes de semillas de distintas variedades capturadas sobre un fondo homogéneo. Este conjunto se encuentra disponible al público para futuras evaluaciones.

Como última contribución se presenta una primera aproximación a la aplicación de los eFV al problema de clasificación de tipos de terreno usando imágenes del tipo PolSAR (capítulo 7). Se propone un modelo que integra los formalismos de eFV con un modelo de energía basado en Potts que captura la dependencia espacial entre las variables. Hasta lo que se conoce, esta es la primera vez que este tipo de codificaciones es aplicada al análisis de imágenes PolSAR y los resultados obtenidos superan a los obtenidos con métodos que son estado del arte. Como contribución relacionada a esta, se puso a disposición un conjunto de anotaciones y la definición de un procedimiento de entrenamiento y evaluación sobre dos conjuntos de datos populares en la literatura. Estos datos y los "scripts" usados en los experimentos se pueden encontrar en la

[8] `http://www.famaf.unc.edu.ar/~jsanchez/efv`

página web del proyecto [9].

Además, durante el desarrollo de la tesis, se publicaron 3 trabajos en congresos nacionales [RSP15b, DTY+15, RGDPC16], 2 en congresos internacionales [RS12, RSP15a] y 3 en revistas indexadas [SR15, DRG+16, RSF17] [10].

1.6. Organización de la tesis

La tesis se puede dividir en tres partes, en la primera, formada por los capítulos 2, 3 y 4 se presenta el estado del arte y se desarrollan los modelos teóricos que son utilizados en los capítulos experimentales. En el capítulo 2 se brinda el marco teórico y contexto que da origen a los modelos y métodos propuestos en la presente tesis. Estos modelos conocidos como FV son extendidos a una más amplia familia de distribuciones de probabilidad (capítulo 3) para crear un nuevo modelo conocido como eFV. En el capítulo 4 se presenta otro modelo de clasificación que ha ganado mucha atención recientemente el cual está basado en Aprendizaje Profundo (DL, por su denominación en inglés). Este último modelo es aplicado como herramienta en capítulos subsiguientes en conjunto con eFV.

En la segunda parte, de tono más experimental, estos modelos son aplicados a problemas puntuales de AP. En el capítulo 5 se aborda el problema de clasificación de especies de plantas, en el capítulo 6 se presenta la clasificación de distintas variedades de semillas de una misma especie y como última aplicación se propone en el capítulo 7 la clasificación de terrenos usando imágenes PolSAR.

En la última parte se presentan las conclusiones de esta tesis y las perspectivas de trabajos a futuro (capítulo 8).

[9] `http://ciii.frc.utn.edu.ar/JavierAndresRedolfi/sartb`

[10] Está última fue aceptada en la revista para su publicación en la próxima edición.

Capítulo 2

Vectores de Fisher

Índice

Uno de los problemas fundamentales de la CV es la construcción de modelos que puedan capturar la información contenida en las imágenes de una manera simple, con bajo costo computacional y de almacenamiento, pero a la vez robusta y con propiedades de generalización a otros entornos.

La clasificación de imágenes consiste en la asignación de uno o múltiples *conceptos* a una determinada imagen o a parte de ella a partir de su contenido semántico. Estos conceptos dependen de la problemática a abordar y normalmente son conocidos como clases, etiquetas o categorías. En esta tesis los conceptos de interés están relacionados con la AP pero los modelos teóricos desarrollados tienen un campo de aplicación que excede a este ámbito en particular.

En la actualidad para la resolución de problemas de CV en general y para clasificación de imágenes en particular se usan principalmente dos modelos, estos son los modelos basados en Bolsa de Palabras Visuales (BoVW, por su denominación en inglés) y los modelos basados en CNN. En este capítulo se habla de una de las generalizaciones de BoVW más utilizadas en la actualidad conocida como FV y luego en el capítulo 3 se presenta una generalización aún más amplia de FV desarrollada en esta tesis, conocida como eFV. Este modelo conforma la base fundamental de los desarrollos teóricos y experimentos de esta tesis, aunque también se utiliza el modelo de redes profundas el cual ha ganado mucha popularidad en los últimos años. Este último se presenta en el capítulo 4.

2.1. Introducción

Por mucho tiempo, la representación de imágenes más popular para clasificación ha sido el modelo de BoVW [CDF+04]. Esta permite obtener una firma o vector que codifica a una imagen basándose en los elementos visuales que más aparecen en la misma. Estos elementos son llamados “palabras visuales” y todos ellos forman algo conocido como “bolsa” o “vocabulario visual”. Este vocabulario es aprendido en forma “offline” sobre un conjunto grande de imágenes utilizando normalmente un algoritmo conocida como K-Means [M+67, Bis06] el cual se encarga

de buscar los K elementos visuales que más se repiten en ese conjunto de imágenes. Este tipo de métodos surge en un primer momento para la clasificación de textos basados en su contenido, de ahí el nombre de palabras y vocabulario [Har54] y luego es adaptado para la clasificación de imágenes.

Para el cómputo de esta representación sobre una imagen, primero se extraen un conjunto de descriptores locales en la misma, como por ejemplo Transformación de Características Invariantes ante Escala (SIFT, por su denominación en inglés) [Low04] y luego a cada uno de estos descriptores se le asigna la palabra más cercana de dicho vocabulario. Después de asignar los descriptores de la imagen a los elementos más cercanos en el vocabulario se cuenta la cantidad de veces que aparece cada elemento del mismo en la imagen, con esto se obtiene un histograma de ocurrencias de palabras el cual puede ser visto como un vector que resume o codifica a la imagen. Luego este histograma puede ser usado por cualquier algoritmo de clasificación como por ejemplo Máquinas de Soporte Vectorial (SVM, por su denominación en inglés) [CV95]. Este método fue utilizado por primera vez en [CDF$^+$04] para clasificar imágenes de siete categorías distintas, mostrando un rendimiento muy superior con respecto a un clasificador bayesiano.

A estos tipos de codificaciones, que asignan una única cuenta para cada descriptor que aparece en la imagen se las conoce como métodos de “asignación dura”. Una de las primeras extensiones a la formulación de BoVW original fue el uso de “asignación suave” la cual consiste en asignar un descriptor a muchas palabras del vocabulario utilizando algún criterio como puede ser la distancia a la misma [FSMST05, PDCB06, VGVSG10]. Otra extensión común es el uso de pirámides espaciales [LSP06] que consiste en dividir la imagen en varias partes teniendo en cuenta su posición espacial y luego computar un histograma para cada una de las divisiones, de esta manera se pueden tener en cuenta diferentes aspectos de la configuración espacial de la imagen, por ejemplo si se divide la imagen al medio con una línea horizontal quedan dos imágenes que representan el “arriba” y el “abajo” de la misma.

Además de estas extensiones, el modelo de BoVW ha sido generalizado para tener en cuenta estadísticas de mayor orden con la introducción de diferentes tipos de codificaciones, entre las que se destacan el FV, el Vector de Descriptores Localmente Agregados (VLAD, por su denominación en inglés) [JDSP10] y el Super Vector (SV) [ZYZH10]. Entre estos, el FV ha mostrado el mejor comportamiento en clasificación [KCZ11, HWWT14] por esto ha sido elegido en esta tesis como la herramienta fundamental y base para los siguientes desarrollos.

El método de BoVW y las diferentes variaciones nombradas han sido utilizados en diversos problemas de AP, por ejemplo en [GJB$^+$14, HKCL14] para la clasificación de hojas de plantas, en [HHH$^+$15] para la clasificación de semillas de arroz, en [VR14] para la clasificación de insectos en plantaciones y en [QPW$^+$17] para contar la cantidad de frutas en árboles de mango, aunque su uso es aún muy incipiente.

En la actualidad esta representación ha sido reemplazada por el esquema de codificación de imágenes conocido como FV [Sán11, SPMV13] el cual ha sido poco explorado para resolver problemas de AP. Esta representación está basada en un método anterior conocido como Núcleo de Fisher (FK, por su denominación en inglés) el cual utiliza funciones de núcleo y fue utilizado por primera vez para clasificar secuencias de ADN en [JH$^+$99]. El FK combina lo mejor de dos mundos, el generativo o probabilístico y el discriminativo, construyendo una función de núcleo desde un modelo generativo de los datos utilizada luego con un esquema discriminativo de clasificación. Este nuevo modelo codifica los descriptores locales computados en una imagen a través de su desviación con respecto a un modelo generativo de los datos. Esta desviación se calcula como el gradiente del logaritmo de la función de verosimilitud de la muestra con respecto a los parámetros del modelo probabilístico. Esta nueva representación vectorial fue utilizada por primera vez en clasificación de imágenes en [PD07] y fue mejorada en [PSM10]. En el caso particular de clasificación de imágenes las muestras corresponden a los descriptores

locales calculados en los parches [1] y el modelo generativo es un Modelo Mezcla de Gaussianas (GMM, por su denominación en inglés) el cual puede ser visto como un "vocabulario visual probabilístico" en analogía con el vocabulario visual del modelo BoVW.

En el resto de este capítulo se introduce el principio del FK y del FV y se describe el procedimiento para su aplicación al problema de clasificación de imágenes. También se presentan un conjunto de normalizaciones que mejoran significativamente la exactitud de clasificación.

2.2. El vector de Fisher

En esta sección se introducen los fundamentos de la representación FV, pero antes se describe el principio del FK para luego mostrar como puede ser adaptado para la clasificación de imágenes.

2.2.1. El núcleo de Fisher

Sea $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ una muestra de N observaciones $\mathbf{x}_i \in \mathcal{X}$ y sea p_λ una función de distribución de probabilidad (pdf, por su denominación en inglés) que modela el proceso de generación de elementos en $\mathcal{X}$. A su vez $\lambda = [\lambda_1, \cdots, \lambda_M]^T \in \mathbb{R}^M$ denota el vector de M parámetros de p_λ.

De la literatura estadística, la *función de score* es conocida como el gradiente del logaritmo de la función de verosimilitud de los datos con respecto a los parámetros del modelo, esto es:

$$G_\lambda(\mathbf{X}) = \nabla_\lambda \log\ p_\lambda(\mathbf{X}), \tag{2.1}$$

donde ∇ es el operador gradiente y $\log\ p_\lambda(\mathbf{X})$ es el logaritmo de la función de verosimilitud de los datos. Para que el modelo sea tratable se supone que las muestras son independientes e idénticamente distribuidas, con lo que nos queda que $\log\ p_\lambda(\mathbf{X}) = \sum_{i=1}^N \log\ p_\lambda(\mathbf{x}_i)$.

Este gradiente describe la contribución de los parámetros individuales al proceso generativo. En otras palabras, describe como los parámetros del modelo generativo p_λ deben ser modificados para ajustarse mejor a los datos $\mathbf{X}$ [JH$^+$99].

Se hace notar que $G_\lambda(\mathbf{X}) \in \mathbb{R}^M$ y por lo tanto la dimensionalidad de $G_\lambda(\mathbf{X})$ solo depende del número de parámetros M en λ y no del tamaño de la muestra N o de la dimensión de los vectores de observación $\mathbf{x}_i$. Esto da la idea de que se pueden clasificar secuencias que tengan diferentes cardinalidades debido a que el vector resultante tiene la misma dimensionalidad.

De la teoría de la geometría de la información [AN00], una familia paramétrica de distribuciones $\mathcal{S} = \{p_\lambda, \lambda \in \Lambda\}$ puede ser considerada como una variedad riemanniana M_Λ con una métrica local dada por la Matriz de Información de Fisher (FIM, por su denominación en inglés) $\mathbf{I}_\lambda \in \mathbb{R}^{M \times M}$, la cual se define como:

$$\mathbf{I}_\lambda = E_{\mathbf{x} \sim p_\lambda}\left[G_\lambda(\mathbf{X}) G_\lambda(\mathbf{X})^T\right] \tag{2.2}$$

en donde $E_{\mathbf{x} \sim p_\lambda}$ representa la esperanza de $\mathbf{x}$ suponiendo una distribución de probabilidad p_λ.

Siguiendo esta observación, en [JH$^+$99] los autores propusieron medir la similitud entre dos muestras $\mathbf{X}_a$ y $\mathbf{X}_b$ tomadas de $\mathcal{X}$ usando el FK:

$$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+,$$

el cual está definido como:

$$K(\mathbf{X}_a, \mathbf{X}_b) \stackrel{\text{def}}{=} G_\lambda(\mathbf{X}_a)^T \mathbf{I}_\lambda^{-1} G_\lambda(\mathbf{X}_b). \tag{2.3}$$

[1] Sub-imágenes, normalmente rectangulares, extraídas de la imagen de referencia.

El FK puede considerarse como una medida de la similitud entre muestras basándose en como ellas afectarían al modelo (en el sentido de máxima verosimilitud) si fueran usadas para actualizar sus parámetros desde λ hacia $\lambda + d\lambda$ a través de la variedad.

Como $\mathbf{I}_\lambda$ es semi-definida positiva (ver demostración en el apéndice A), su inversa también lo es, por lo tanto usando la descomposición de Cholesky, la matriz inversa de la FIM puede ser escrita como:

$$\mathbf{I}_\lambda^{-1} = \mathbf{L}_\lambda^T \mathbf{L}_\lambda, \tag{2.4}$$

y el núcleo de la ecuación (2.3) puede ser reescrito explícitamente como un producto punto:

$$K(\mathbf{X}_a, \mathbf{X}_b) = \mathcal{G}_\lambda(\mathbf{X}_a)^T \mathcal{G}_\lambda(\mathbf{X}_b), \tag{2.5}$$

donde

$$\mathcal{G}_\lambda(\mathbf{X}) = \mathbf{L}_\lambda G_\lambda(\mathbf{X}) = \mathbf{L}_\lambda \nabla_\lambda \log\ p_\lambda(\mathbf{X}). \tag{2.6}$$

Este vector gradiente, normalizado con la FIM, es llamado el FV de $\mathbf{X}$ y una máquina de núcleo no lineal usando la ecuación (2.5) como núcleo es equivalente a una máquina de núcleo lineal usando $\mathcal{G}_\lambda(\mathbf{X})$ como vector de características.

Un beneficio claro de esta nueva formulación es que los FV pueden ser utilizados con clasificadores lineales los cuales pueden ser entrenados en forma muy eficiente en comparación con clasificadores no lineales. Por ejemplo como se propone en [HKS11], donde se presenta un método de aprendizaje basado en descenso de gradiente estocástico el cual tiene una dependencia sublineal del tiempo de entrenamiento con respecto a la cantidad de muestras de entrenamiento.

2.3. Uso en clasificación de imágenes

En esta sección se establecen cuales son los pasos necesarios para computar la codificación FV en imágenes para su uso en clasificación.

2.3.1. Descriptores

Para aplicar este modelo lo primero a hacer es definir cuales serán las observaciones utilizadas, que para el caso de imágenes son un conjunto de descriptores locales, por ejemplo un conjunto de descriptores SIFT [Low04]. Para obtener este conjunto se deben computar estos descriptores en distintas partes de la imagen. Existen diferentes formas de elegir las ubicaciones en las que se computarán los descriptores, ya sea a través de detectores de características para seleccionar las regiones más salientes de la imagen o por ejemplo en forma aleatoria. Lo normal para este tipo de codificaciones es usar una grilla de muestreo regular, también conocido como muestreo denso. En la figura 2.1 se muestra un ejemplo de las ubicaciones en las que se calculan los descriptores locales; sobre cada cuadrado o parche de la grilla se calcula un descriptor. En el ejemplo mostrado no hay superposición entre los parches pero puede ocurrir que el paso sea menor que el tamaño de los parches por lo tanto los parches se superponen y dejan de ser independientes entre sí (ver figura 2.2).

Además de esto, se utilizan pirámides de resolución lo cual consiste en calcular los descriptores en forma densa sobre la imagen original y en varias versiones escaladas de la misma. Normalmente se utiliza un escalado de $\frac{1}{\sqrt{2}}$ y se computan descriptores sobre cuatro o cinco escalas. En la figura 2.3 se muestra una imagen en conjunto con cuatro escalas de la misma.

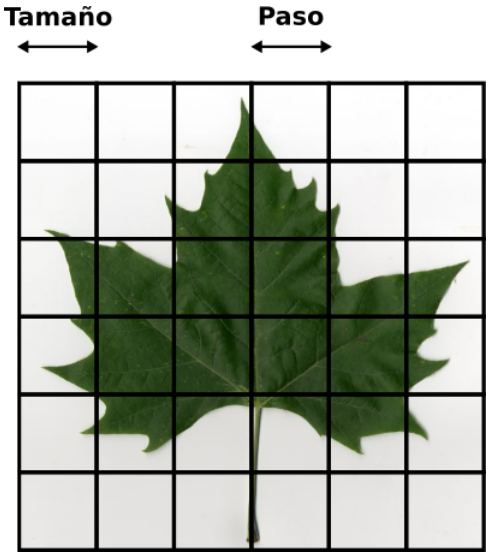


Figura 2.1: Ejemplo de una grilla densa superpuesta sobre una imagen para el cálculo de descriptores locales. Sobre cada cuadrado se computa un descriptor. En este caso el paso y el tamaño coinciden, por ello no hay superposición entre parches.

2.3.1.1. Análisis de componentes principales

Un paso clave para el buen funcionamiento de los FV es aplicar un procedimiento de reducción de dimensionalidad [SPMV13] a los descriptores. El algoritmo usado para esto es conocido como Análisis de Componentes Principales (PCA, por su denominación en inglés) y lo que permite es proyectar ortogonalmente un conjunto de descriptores a un subespacio de menor dimensionalidad, conocido como subespacio principal [Bis06], en el cual se maximiza la varianza de los datos proyectados pero se reducen las covarianzas. Esto último hace que los datos proyectados se ajusten mejor a la suposición de matrices de covarianza diagonales que se realizará más adelante. El cómputo de la matriz de proyección de los datos se realiza sobre un subconjunto de descriptores tomados en forma aleatoria desde las imágenes en consideración.

2.3.2. Codificación

Para simplificar el problema se asume que los descriptores (muestras) están distribuidas en forma idéntica e independiente. Aunque esta asunción es el procedimiento normal en los modelos del tipo BoVW, se hace notar que esta no es del todo cierta en la realidad ya que es más problable que descriptores tomados en la misma región contengan la misma información o información relacionada y más aún si estos son muestreados con solapamiento. En la sección 2.3.8 se presenta una normalización que reduce el efecto de esta incorrecta asunción.

Llámese ahora $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, a ese conjunto de descriptores locales extraídos de la imagen donde cada $\mathbf{x}_i \in \mathbb{R}^D$. Asumiendo que las muestras son i.i.d, se puede reescribir la ecuación (2.6) como sigue:

$$\mathcal{G}_\lambda(\mathbf{X}) = \sum_{i=1}^{N} \mathbf{L}_\lambda \nabla_\lambda \log\ p_\lambda(\mathbf{x}_i). \tag{2.7}$$

Por lo tanto, bajo esta suposición de independencia, el FV es la estadística de una suma norma-

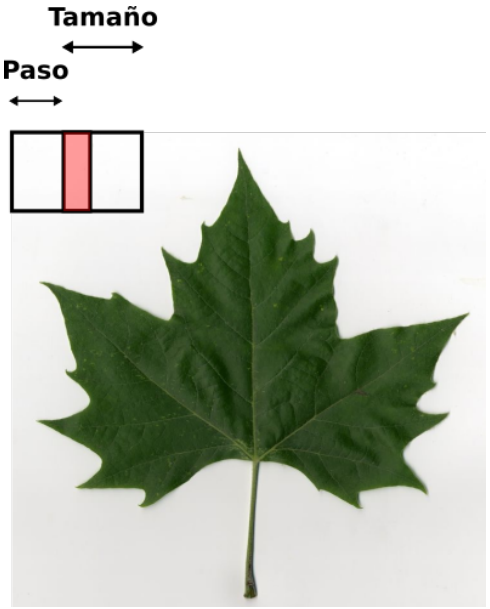


Figura 2.2: Ejemplo de muestreo cuando el paso es menor que el tamaño del descriptor. En rojo claro se muestra el área de solapamiento.

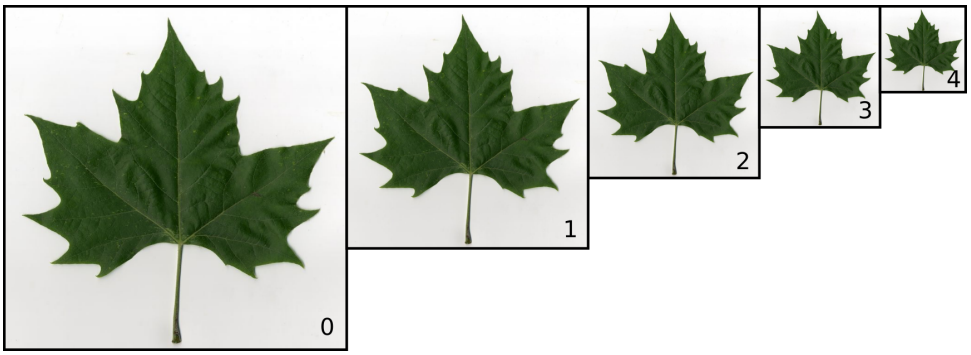


Figura 2.3: Pirámides de resolución. En este caso se generan 4 escalas de la imagen original y los descriptores se calculan en forma densa sobre la imagen original y las 4 imágenes escaladas.

lizada de gradientes $\mathbf{L}_\lambda \nabla_\lambda \log\ p_\lambda(\mathbf{x}_i)$ computada para cada descriptor. La operación:

$$\mathbf{x}_i \to \varphi_{\mathrm{FV}}(\mathbf{x}_i) = \mathbf{L}_\lambda \nabla_\lambda \log\ p_\lambda(\mathbf{x}_i). \tag{2.8}$$

puede ser entendida como la proyección del descriptor local $\mathbf{x}_i$ en un espacio de mayor dimensionalidad. Dicho espacio es más ameno para la clasificación lineal debido a que en el es más simple encontrar un hiper-plano que separe a las muestras. Nótese que la presunción de independencia de los parches en una imagen es generalmente incorrecta [CVS12], especialmente cuando los parches se superponen como se muestra en la figura 2.2.

2.3.3. Modelo probabilístico

De aquí en adelante, se elige p_λ como un GMM. En la literatura de visión por computadora, un GMM que modela el proceso de generación de descriptores locales en cualquier imagen es

referido como un vocabulario visual universal (probabilístico) [PDCB06, WCM05], en analogía con el vocabulario visual del modelo BoVW.

En la ecuación (2.10) se puede ver la definición del modelo GMM, en donde los parámetros se denotan con:

$$\lambda = \{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, k = 1, \cdots, K\}, \tag{2.9}$$

donde $w_k \in \mathbb{R}_+$, $\boldsymbol{\mu}_k \in \mathbb{R}^D$ y $\boldsymbol{\Sigma}_k \in \mathbb{R}^{D\times D}$ son respectivamente el peso, el vector de medias y la matriz de covarianza de la gaussiana K. Este modelo se escribe como:

$$p_\lambda(\mathbf{x}) = \sum_{k=1}^{K} w_k p_k(\mathbf{x}), \tag{2.10}$$

donde $p_k : \mathcal{X} \to \mathbb{R}_+$ denota a la gaussiana k, la cual se define como:

$$p_k(\mathbf{x}) = \frac{1}{(2\pi)^{D/2} \mid \boldsymbol{\Sigma}_k \mid^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right], \tag{2.11}$$

requiriendo que:

$$\forall_k : w_k \geq 0, \qquad \sum_{k=1}^{K} w_k = 1, \tag{2.12}$$

para asegurar que p_λ sea una distribución válida. En lo que sigue, se asumen matrices de covarianza diagonales lo cual es una procedimiento estándar y se denotará con $\boldsymbol{\sigma}_k^2$ al vector de varianzas, es decir la diagonal de $\boldsymbol{\Sigma}_k$.

Para los pesos se adopta el formalismo de reparametrización propuesto en [KVJ11] que plantea expresar los pesos w_k en función de otros parámetros α_k:

$$w_k = \frac{\exp(\alpha_k)}{\sum_{j=1}^{K} \exp(\alpha_j)} \tag{2.13}$$

Esta reparametrización usando α_k evita forzar explícitamente las restricciones de la ecuación (2.12) lo cual simplifica el cálculo de los gradientes.

2.3.4. Fórmulas del gradiente

Para el cómputo del FV primero se necesita calcular los gradientes con respecto a los parámetros de la ecuación (2.7) para luego normalizarlos con la FIM y por último concatenarlos para generar el vector.

Los gradientes de un descriptor $\mathbf{x}_i$ con respecto a los parámetros λ de la GMM son (ver apéndice B para el cálculo de los mismos):

$$\nabla_{\alpha_k} \log\ p_\lambda(\mathbf{x}_i) = \gamma_k(\mathbf{x}_i) - w_k, \tag{2.14}$$

$$\nabla_{\boldsymbol{\mu}_k} \log\ p_\lambda(\mathbf{x}_i) = \gamma_k(\mathbf{x}_i)\left(\frac{\mathbf{x}_i - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k^2}\right), \tag{2.15}$$

$$\nabla_{\boldsymbol{\sigma}_k} \log\ p_\lambda(\mathbf{x}_i) = \gamma_k(\mathbf{x}_i)\left[\frac{(\mathbf{x}_i - \boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^3} - \frac{1}{\boldsymbol{\sigma}_k}\right], \tag{2.16}$$

donde $\gamma_k(\mathbf{x}_i)$ es la asignación liviana o suave de $\mathbf{x}_i$ a la gaussiana k. Esta asignación también es

conocida como probabilidad a posterior o responsabilidad [Bis06]:

$$\gamma_k(\mathbf{x}_i) = \frac{w_k p_k(\mathbf{x}_i)}{\sum_{j=1}^{K} w_j p_j(\mathbf{x}_i)}, \tag{2.17}$$

en donde las operaciones de división y exponenciación de vectores deben ser entendidas como operaciones término a término.

2.3.5. Cálculo de la FIM

Teniendo una expresión para los gradientes, la cuestión restante es como computar $\mathbf{L}_\lambda$. En [JH+99] se propone usar como FIM una matriz identidad, esto es lo mismo que no usar esta normalización. Según los autores la presencia de la FIM es poco significativa y el uso del núcleo simplificado $K_S(\mathbf{X}_a, \mathbf{X}_b) = G_\lambda(\mathbf{X}_a)^T G_\lambda(\mathbf{X}_b)$ es una elección apropiada, además con esto se evita el cómputo e inversión de esta matriz lo cual es muy costoso principalmente por la dimensionalidad de la misma. Posteriormente, en [PD07] se propone aproximarla usando una forma diagonal, esto consiste en suponer que todas las componentes de la matriz son cero, excepto sus valores diagonales. Una ventaja de esta aproximación es que permite obtener una forma cerrada para los valores diagonales. Luego en [Sán11, SPMV13] se demuestra que la aproximación diagonal surge de manera natural suponiendo que existe una asignación dura de descriptores a las palabras del vocabulario. Esta aproximación es la usada en esta tesis.

La FIM diagonal, ahora puede ser vista como un vector $\mathbf{I}_{D_\lambda}$, en donde cada componente j del vector se corresponde con la componente (j, j) de la matriz $\mathbf{I}_\lambda$. El vector $\mathbf{I}_{D_\lambda}$ queda de la siguiente forma:

$$\mathbf{I}_{D_\lambda} = [f_{\alpha_0}, \cdots, f_{\alpha_K}, f_{\boldsymbol{\mu}_0}, \cdots, f_{\boldsymbol{\mu}_K}, f_{\boldsymbol{\sigma}_0}, \cdots, f_{\boldsymbol{\sigma}_K}]^T \tag{2.18}$$

en donde:

$$\begin{aligned} f_{\alpha_k} &= \frac{\partial^2}{\partial \alpha_k^2}[G_\lambda(\mathbf{X})] \approx w_k \\ f_{\boldsymbol{\mu}_k} &= \frac{\partial^2}{\partial \boldsymbol{\mu}_k^2}[G_\lambda(\mathbf{X})] \approx \frac{w_k}{\boldsymbol{\sigma}_k^2} \\ f_{\boldsymbol{\sigma}_k} &= \frac{\partial^2}{\partial \boldsymbol{\sigma}_k^2}[G_\lambda(\mathbf{X})] \approx \frac{2w_k}{\boldsymbol{\sigma}_k^2}, \end{aligned}$$

para $k = 1, \cdots, K$ (para la derivación de estas aproximaciones ver el apéndice del trabajo [SPMV13]). Por último para obtener la matriz de normalización $\mathbf{L}_\lambda$ se usa la ecuación (2.4) y se obtiene:

$$\mathbf{L}_{D_\lambda} = [l_{\alpha_0}, \cdots, l_{\alpha_K}, l_{\boldsymbol{\mu}_0}, \cdots, l_{\boldsymbol{\mu}_K}, l_{\boldsymbol{\sigma}_0}, \cdots, l_{\boldsymbol{\sigma}_K}]^T, \tag{2.19}$$

en donde:

$$l_{\alpha_k} \approx \frac{1}{\sqrt{w_k}}, \qquad l_{\boldsymbol{\mu}_k} \approx \frac{\boldsymbol{\sigma}_k}{\sqrt{w_k}}, \qquad l_{\boldsymbol{\sigma}_k} \approx \frac{\boldsymbol{\sigma}_k}{\sqrt{2w_k}},$$

para $k = 1, \cdots, K$.

2.3.6. Armado del FV

Una vez computado el vector $\mathbf{L}_{D_\lambda}$, este es usado para la normalización coordenada a coordenada de los vectores gradiente (ecuación (2.7)) obteniendo los siguientes gradientes norma-

lizados:

$$\mathcal{G}_{\alpha_k}(\mathbf{X}) = \frac{1}{\sqrt{w_k}} \sum_{i=1}^{N} [\gamma_k(\mathbf{x}_i) - w_k], \tag{2.20}$$

$$\mathcal{G}_{\boldsymbol{\mu}_k}(\mathbf{X}) = \frac{1}{\sqrt{w_k}} \sum_{i=1}^{N} \gamma_k(\mathbf{x}_i) \left(\frac{\mathbf{x}_i - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k} \right), \tag{2.21}$$

$$\mathcal{G}_{\boldsymbol{\sigma}_k}(\mathbf{X}) = \frac{1}{\sqrt{w_k}} \sum_{i=1}^{N} \gamma_k(\mathbf{x}_i) \frac{1}{\sqrt{2}} \left[\frac{(\mathbf{x}_i - \boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^2} - 1 \right], \tag{2.22}$$

Se aclara nuevamente que en estas ecuaciones las operaciones de división y exponenciación entre vectores se realizan término a término. Notar que $\mathcal{G}_{\alpha_k}(\mathbf{X})$ es un escalar mientras que $\mathcal{G}_{\boldsymbol{\mu}_k}(\mathbf{X})$ y $\mathcal{G}_{\boldsymbol{\sigma}_k}(\mathbf{X})$ son vectores de D dimensiones. El FV final es la concatenación de los gradientes $\mathcal{G}_{\alpha_k}(\mathbf{X})$, $\mathcal{G}_{\boldsymbol{\mu}_k}(\mathbf{X})$ y $\mathcal{G}_{\boldsymbol{\sigma}_k}(\mathbf{X})$ para $k = 1, \cdots, K$:

$$\mathcal{G}_{\lambda}(\mathbf{X}) = [\mathcal{G}_{\alpha_0}(\mathbf{X}), \cdots, \mathcal{G}_{\alpha_K}(\mathbf{X}), \mathcal{G}_{\boldsymbol{\mu}_0}(\mathbf{X}), \cdots, \mathcal{G}_{\boldsymbol{\mu}_K}(\mathbf{X}), \mathcal{G}_{\boldsymbol{\sigma}_0}(\mathbf{X}), \cdots, \mathcal{G}_{\boldsymbol{\sigma}_K}(\mathbf{X})]^T, \tag{2.23}$$

y por lo tanto su dimensionalidad es $E = (2D + 1)K$.

Para evitar la dependencia del tamaño de la muestra se normaliza el FV resultante por el tamaño de la muestra N, es decir que se realiza la siguiente operación [2]:

$$\mathcal{G}_{\lambda}(\mathbf{X}) \leftarrow \frac{1}{N} \mathcal{G}_{\lambda}(\mathbf{X}) \tag{2.24}$$

Las ecuaciones (2.20), (2.21) y (2.22) pueden ser computadas en términos de las siguientes estadísticas de orden 0, 1 y 2:

$$S_k^0 = \sum_{i=1}^{N} \gamma_k(\mathbf{x}_i), \tag{2.25}$$

$$S_k^1 = \sum_{i=1}^{N} \gamma_k(\mathbf{x}_i) \mathbf{x}_i, \tag{2.26}$$

$$S_k^2 = \sum_{i=1}^{N} \gamma_k(\mathbf{x}_i) \mathbf{x}_i^2, \tag{2.27}$$

donde $S_k^0 \in \mathbb{R}$, $S_k^1 \in \mathbb{R}^D$ y $S_k^2 \in \mathbb{R}^D$. Como antes, el cuadrado de un vector debe ser entendido como una operación término a término.

2.3.7. Estimación de los parámetros del modelo.

La distribución dada por la ecuación (2.10) es un modelo para la generación de muestras en cualquier imagen, sin importar a que clase pertenezcan. Sus parámetros deben, por lo tanto, ser estimados basándose en un conjunto diverso de elementos muestreados en forma aleatoria desde las imágenes en consideración. Sea $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ tal conjunto; entonces para estimar los parámetros del modelo se resuelve una formulación de máxima verosimilitud de los datos a través del algoritmo de Esperanza-Maximización (EM) [DLR77]. Dicho algoritmo es un procedimiento del tipo iterativo que tiene los siguientes pasos (para un tratamiento con mayor

[2] Esta normalización es una heurística que funciona en la práctica. En la sección 3.4 se generaliza la definición de FV sobre conjuntos y se deriva una solución que considera el tamaño de la muestra.

profundidad ver [Bis06]):

1. *Inicialización*: inicializar los parámetros de la ecuación (2.9):

$$\lambda = \{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, k = 1, \cdots, K\},$$

2. ***E**speranza*: computar la asignación liviana de muestras a componentes de la mezcla para cada elemento del conjunto de entrenamiento, ecuación (2.17):

$$\gamma_k(\mathbf{x}_i) = \frac{w_k p_k(\mathbf{x}_i)}{\sum_{j=1}^{K} w_j p_j(\mathbf{x}_i)},$$

   $k = 1, \ldots, K$ y $i = 1, \ldots, N$, usando las estimaciones actuales de los parámetros.

3. ***M**aximización*: actualizar los parámetros del modelo usando los posterioris computados en el paso anterior. Las ecuaciones de actualización adoptan la siguiente forma:

$$\begin{aligned}
\boldsymbol{\mu}_k^{(t+1)} &\leftarrow \frac{1}{N_k} \sum_{i=1}^{N} \gamma_k^{(t)}(\mathbf{x}_i)\mathbf{x}_i, \\
\boldsymbol{\sigma}_k^{(t+1)} &\leftarrow \frac{1}{N_k} \sum_{i=1}^{N} \gamma_k^{(t)}(\mathbf{x}_i)\left(\mathbf{x}_i - \boldsymbol{\mu}_k^{(t)}\right)^2, \\
\alpha_k^{(t+1)} &\leftarrow \frac{N_k}{N}, \\
w_k^{(t+1)} &\leftarrow \frac{\exp\left(\alpha_k^{(t)}\right)}{\sum_{j=1}^{K} \exp\left(\alpha_j^{(t)}\right)}
\end{aligned}$$

   donde $N_k = \sum_{i=1}^{N} \gamma_k^{(t)}(\mathbf{x}_i)$, para $k = 1, \ldots, K$ (nuevamente las operaciones vectoriales se entienden término a término).

4. *Convergencia*: calcular el logaritmo de la función de verosimilitud de los datos y compararla con el valor obtenido en el paso anterior:

$$\log\ p_\lambda(\mathbf{X}) = \sum_{i=1}^{N} \log\ p_\lambda(\mathbf{x}_i)$$

Estos pasos son repetidos en forma iterativa hasta que el valor esperado completo de los datos deje de mejorar o hasta que se alcance un número máximo de iteraciones. La inicialización del algoritmo puede ser realizada de diferentes maneras, por ejemplo muestreando K prototipos en forma aleatoria desde el conjunto de entrenamiento o ejecutando algún algoritmo de agrupamiento sobre un subconjunto de los datos [SJN13] como K-Means.

2.3.8. Normalización de los FV

Por último se describen dos pasos de normalización que fueron introducidos en [PSM10] los cuales son necesarios para obtener resultados competitivos cuando los FV son combinados con clasificadores lineales. Para un tratamiento más profundo sobre ambas normalizaciones y sus justificaciones teóricas ver [SPMV13].

Normalización ℓ_2 En [PSM10] se propone la normalización ℓ_2 de los FV. Esta normalización consiste en dividir al vector por el módulo del mismo como se muestra a continuación:

$$\mathbf{z} \leftarrow \frac{\mathbf{z}}{\parallel \mathbf{z} \parallel} \tag{2.28}$$

La interpretación más común de esta normalización, propuesta en [PSM10] plantea que la normalización ℓ_2 está justificada como una manera de cancelar el hecho de que imágenes distintas contienen diferentes cantidades de información de fondo. Con ella se logra que la información independiente de la imagen (información común a todas las imágenes) sea casi descartada del FV, una propiedad deseable. Sin embargo, el FV sigue siendo dependiente de la proporción de información específica de la imagen después de la normalización. Esta normalización, además permite que el producto punto entre vectores sea una medida de similitud entre ellos, comúnmente conocida distancia coseno.

Normalización de potencia En [PSM10], se propone realizar una normalización de potencia de la forma:

$$z \leftarrow \text{signo}(z) \mid z \mid^{\phi} \quad \text{con} \quad 0 < \phi \leq 1 \tag{2.29}$$

a cada dimensión del FV en donde la función signo(z) es igual a $+1$ si $z >= 0$ e igual a -1 en caso contrario y $\mid z \mid$ es el valor absoluto de z. La elección normal de este coeficiente es $\phi = 0{,}5$, razón por la cual esta transformación también es conocida como raíz cuadrada con signo.

En la literatura existen muchas explicaciones para justificar tal transformación. En [PSM10] se argumenta que, como el número de gaussianas del modelo GMM aumenta, el FV se vuelve ralo lo cual impacta sobre el producto punto. En el caso donde los FV son extraídos de subregiones, el efecto de picos en ciertas partes del vector se vuelve mas evidente ya que el número de descriptores agregados es menor en comparación con los extraídos en toda la imagen. La normalización de potencia suaviza este efecto de pico mejorando el comportamiento del producto punto como medida de similitud. Otra interpretación propuesta en [PLSP10] es que reduce la influencia de descriptores que ocurren frecuentemente dentro de una imagen, de manera similar a lo que proponen en [JDS09]. En otras palabras, la raíz cuadrada corrige por la incorrecta presunción de independencia realizada en la ecuación 2.7.

2.3.9. Resumen del uso de FV para la clasificación de imágenes

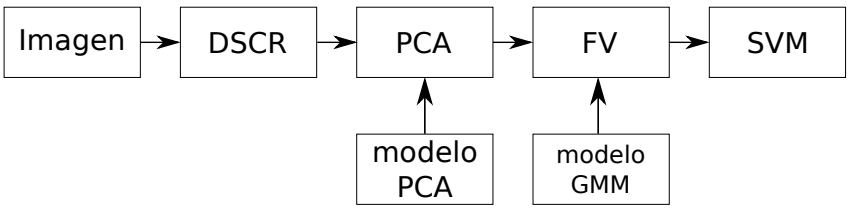


Figura 2.4: Diagrama en bloques del procedimiento para el uso de los FV en clasificación de imágenes.

En la figura 2.4 se muestra un diagrama en bloques del procedimiento de cómputo de la codificación FV. Antes de realizar la codificación se deben entrenar dos modelos, el primero es la matriz de reducción de dimensionalidad PCA y el segundo es el modelo probabilístico GMM. Una vez computados estos modelos, la entrada del algoritmo es la imagen sobre la cual se calculan los descriptores locales en forma densa (ver figura 2.1). Luego se aplica la transformación de reducción de dimensionalidades y sobre estos descriptores se aplica la codificación FV. Una

vez obtenidos los vectores se debe entrenar un clasificador, en este caso un SVM lineal, usando los FV obtenidos de las imágenes de entrenamiento. Por último, este modelo se utiliza para clasificar un nuevo vector computado sobre una nueva imagen de evaluación.

Capítulo 3

Vectores de Fisher de la familia exponencial

Índice

3.1. Resumen

Uno de los problemas fundamentales en clasificación de imágenes es idear modelos que permitan relacionar imágenes con conceptos semánticos de alto nivel de una manera eficiente y confiable. Un enfoque ampliamente usado consiste en la extracción de descriptores locales de las imágenes para luego resumirlos en una representación a nivel de imagen. Dentro de este enfoque, vectores de Fisher es una de las codificaciones más robustas a la fecha. En la codificación FV, los descriptores locales son modelados como muestras sacadas de una mezcla de funciones de distribución de probabilidad gaussianas. Luego una imagen está representada por el vector gradiente que caracteriza las distribuciones de muestras con respecto al modelo. Equipadas con características poderosas como SIFT, FV ha mostrado resultados de estado del arte en diferentes problemas de reconocimiento. Sin embargo, no está claro como se puede aplicar esto cuando el espacio de características es claramente no Euclidiano, conduciendo a heurísticas que ignoran la estructura subyacente del espacio. En este capítulo se generalizan los FV gaussianos a una más amplia familia de distribuciones conocida como la *familia exponencial*. El modelo, bautizado *vectores de Fisher de la familia exponencial* (eFV), provee un marco teórico unificado desde el

cual se pueden derivar representaciones ricas y poderosas. Resultados experimentales muestran la generalidad y flexibilidad de este enfoque.

3.2. Introducción

Este capítulo, nuevamente se enfoca en el problema de clasificación de imágenes, esto es la tarea de asignar etiquetas a imágenes basándose en su contenido. Motivado por el tremendo crecimiento en el volumen y la complejidad de los datos relacionados con las imágenes, el problema ha atraído gran interés. Actualmente, no solo el número de imágenes ha crecido sino que también la naturaleza de la información visual está cambiando hacia modalidades más complejas, por ejemplo el uso de información de profundidad por la aparición de las cámaras RGBD [1] [WLW+14, GGAM14], el reciente interés en imágenes multiespectrales [SLCS14] y el uso de imágenes satelitales como por ejemplo las PolSAR [LGA+99] para resolver diferentes tipos de problemas de percepción. Idear métodos que permitan capturar la rica información semántica codificada en las imágenes se mantiene como un asunto importante.

Como se dijo en el capítulo anterior, uno de los enfoques más exitosos para abordar el problema ha sido representar las imágenes con un resumen estadístico computado desde un conjunto de descriptores de parches locales y usar estas "firmas" para aprender los clasificadores. El ejemplo más emblemático de estos modelos ha sido BoVW [CDF+04, SZ03]. En BoVW, los descriptores locales son primero codificados en vectores de largo fijo usando una representación auxiliar conocida como código visual. Luego estos vectores son agregados en una representación global a través de una operación de agrupamiento (promedio para este caso) y usados como entrada a un clasificador.

Una presunción subyacente de este modelo y las generalizaciones nombradas en el capítulo anterior es que los descriptores locales están, al menos localmente, normalmente [2] distribuidos. Para BoVW, VLAD y SV esto es motivado por el uso de distancia euclidea durante el paso de codificación mientras que en FV esto sigue de la elección explícita de una mezcla de pdf gaussianas (GMM) para modelar la distribución de características locales. A pesar del gran suceso de estos modelos cuando se combinan con descriptores robustos como SIFT [Low04], no está claro como deben ser aplicados en casos en donde el espacio de descriptores locales es claramente no gaussiano, por ejemplo para el caso de descriptores binarios [CLO+12, AOV12] o cuando están definidos sobre el espacio de matrices simétricas de $n \times n$ positivas definidas [TPM06, MSJ14]. Notar que esta observación también se mantiene para espacios de características que son subconjuntos de $\mathbb{R}^n$, por ejemplo histogramas normalizados en el símplex $(n-1)$ estándar de características locales proyectadas en la esfera unitaria a través de una operación de normalización. Cuando se tienen que manejar estas situaciones en la práctica, es común pre o post-procesar los datos para ajustarlos un poco más a las suposiciones realizadas en el modelo, y de esta manera la formulación principal se mantiene sin modificaciones. Un ejemplo muy común de esta estrategia es el paso de preprocesamiento realizado aplicando PCA ampliamente usado en la codificación de descriptores SIFT con FV [SPMV13].

Aunque efectivo en la práctica, estas heurísticas no son muy satisfactorias desde el punto de vista del modelado ya que ellas ignoran la estructura subyacente de las datos. Como ejemplo ilustrativo, se pueden considerar descriptores binarios como los propuestos en [CLO+12, AOV12]. Esta familia de descriptores goza de varias propiedades que los hacen muy atractivos para problemas de reconocimiento de gran escala, por ejemplo son muy rápidos de computar (órdenes de magnitud más rápidos que SIFT) y ocupan mucha menos memoria que su contraparte real. Sin embargo, hasta ahora han sido restringidos solo a problemas de corresponden-

[1] Cámaras con 4 canales, los 3 primeros se corresponden los colores y el último mide distancia.

[2] Con "normalmente" se hace referencia a que siguen una distribución gaussiana o normal.

cia [HDF12] y problemas de reconocimiento a nivel de instancias.

Uno de los primeros intentos de usar descriptores binarios modernos en problemas de reconocimiento de alto nivel puede ser encontrado en [GLT11], en donde los autores proponen aprender un modelo de bolsa de palabras binarias (BoBW) usando K-Means estándar seguido por una operación de redondeo sobre los elementos del código. Usando un enfoque más justificado, en [ZZBC13] proponen un esquema de aprendizaje basado en la distancia de Hamming que probó ser útil en clasificación. Más relacionado con la propuesta de esta tesis, en [USS16] los autores derivaron un FV basado en mezclas de pdfs de Bernoulli las cuales han mostrado un mejor comportamiento que BoVW en tareas de recuperación de objetos en imágenes. En [CAGA14], los autores proponen un modelo que extiende BoVW computando histogramas de distancia entre el conjunto de descriptores y cada elemento en el código, el cual fue aprendido usando el algoritmo de k-medianas y distancia de Hamming.

Más allá del caso de descriptores binarios, ha habido un creciente interés en el uso de matrices de covarianza como descriptores locales. Pero, como las matrices de covarianza yacen en una variedad bastante compleja, usarlas en forma apropiada es bastante desafiante. En esta línea de trabajo, en [TPM06] se consideró el uso de descriptores de covarianza construidos desde características simples computadas a nivel de pixel (incluyendo la posición, información de color y primera y segunda derivada espacial). Para la clasificación, se basaron en un esquema de aumentado (*boosting*) usando clasificadores de K Vecinos más Cercanos y una métrica de distancia especializada para matrices de covarianza. En el contexto de análisis de formas 3D, [TLPG14] propuso un modelo que extiende BoVW con el uso de distancias geodésicas en la variedad de matrices Simétricas Positiva Definida (SPD). El enfoque muestra una exactitud superior en la correspondencia de formas y en tareas de recuperación de imágenes comparado con otros enfoques basados en descriptores. Con el mismo espíritu, en [FHWL14] se propone un modelo de "bolsa de palabras" riemannianas basándose en la media de Karcher [Pen06] (aprendizaje de código) y la divergencia de Stein [Sra11] (asignación de muestras). En el mismo trabajo, los autores proponen un modelo de "tensor de Fisher" el cual consiste en un embebido de la variedad de matrices SPD en un espacio vectorial en el cual puedan ser aprendidos FV gaussianos. Estos modelos fueron satisfactoriamente aplicados para la clasificación de células humanas desde imágenes 2D.

En este capítulo, se presenta una generalización del formalismo de FV a una amplia familia de distribuciones conocida como *familia exponencial*. Ya que los miembros de esta familia están definidos sobre una amplia variedad de dominios, el modelo, nombrado eFV, provee un entorno unificado desde el cual pueden ser derivadas representaciones flexibles y poderosas.

Las principales contribuciones de este capítulo son las siguientes. Se presenta una derivación completa de FV sobre conjuntos, considerando también el caso en que varía la cardinalidad de la muestra (Sección 3.3). Se extiende la normalización de diagonalización en la formulación original a una forma diagonal en bloques y se provee un método simple y general para su estimación. Se analizan los casos de espacios de entrada finitos y se demuestra que, en este caso, la clasificación lineal se vuelve independiente de la complejidad del modelo (Sección 3.6.1). Se muestra sobre dos problemas de clasificación diferentes y desafiantes (Sección 3.7) el poder y flexibilidad del enfoque propuesto. Además el código usado para entrenar los modelos y para computar la codificación eFV se empaquetó en una librería que se llamó **vrl** la cual encuentra disponible para descarga en la página web del proyecto (`http://www.famaf.unc.edu.ar/~jsanchez/efv`) y es utilizada en el resto de los capítulos experimentales de la tesis.

El contenido de este capítulo está basado fundamentalmente en el trabajo [SR15] el cual fue publicado en la revista Pattern Recognition Letters.

3.3. El vector de Fisher

Como ya se dijo en el capítulo anterior, el FK entre dos muestras $\mathbf{X}_a$ y $\mathbf{X}_b$ tomadas desde $\mathcal{X}$ está definido como:

$$K(\mathbf{X}_a, \mathbf{X}_b) \stackrel{\text{def}}{=} [\mathbf{L}_\lambda \nabla_\lambda \log p_\lambda(\mathbf{X}_a)]^T [\mathbf{L}_\lambda \nabla_\lambda \log p_\lambda(\mathbf{X}_b)] = \mathcal{G}_\lambda^T(\mathbf{X}_\mathrm{a}) \mathcal{G}_\lambda(\mathbf{X}_\mathrm{b}),$$

en donde el mapeo $\mathcal{G}_\lambda : \mathcal{X} \to \mathbb{R}^M$ está definido como:

$$\mathcal{G}_\lambda(\mathbf{X}) \stackrel{\text{def}}{=} \mathbf{L}_\lambda \nabla_\lambda \log p_\lambda(\mathbf{X})) \tag{3.1}$$

y es conocido como la codificación FV de $\mathbf{X}$.

3.4. Vectores de Fisher sobre conjuntos

Hágase que $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ sea un conjunto de N muestras i.i.d tomadas desde $\mathcal{X}$ y supóngase que p_λ es una distribución válida sobre $\mathcal{X}$. Analizando los valores de N, se pueden tener dos casos, de acuerdo a si N puede ser considerado como una constante o si depende de cada $\mathbf{X}$ particular. En el primer caso, el cual se da normalmente cuando las imágenes que se analizan tienen el mismo tamaño o se redimensionan para que esto suceda, la codificación FV de cualquier $\mathbf{X}$ dado, puede ser escrita como:

$$\mathcal{G}_\lambda(\mathbf{X}) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} \mathbf{L}_\lambda \nabla_\lambda \log p_\lambda(\mathbf{x}_i) \ . \tag{3.2}$$

El factor $1/\sqrt{N}$ resulta de la descomposición de la FIM para la distribución producto $\prod_{i=1}^N p_\lambda(\mathbf{x}_i)$ usando el hecho de que la FIM de N experimentos independiente es la suma de la información de cada experimento calculada en forma independiente. En este caso el producto punto entre el FV de $\mathbf{X}_a$ y $\mathbf{X}_b$ es el FK entre las muestras. Sin embargo, cuando N es variable, el producto punto entre las codificaciones generadas por (3.2) no corresponde más con la descomposición explícita del FK como sí lo hacía antes.

A pesar de eso, se puede extender la formulación de arriba introduciendo la cardinalidad de las muestras en forma explícita en el modelo de la siguiente manera: defínase $N = \mathrm{card}\,(\mathbf{X})$ como una variable aleatoria con una distribución de Poisson (ver apéndice C) con parámetro θ y considérese el siguiente modelo conjunto para N y $\mathbf{X}$:

$$p_{(\theta,\lambda)}(\mathbf{X}, N) = p_\theta(N) p_\lambda(\mathbf{X}|N) = p_\theta(N) \prod_{i=1}^{N} p_\lambda(\mathbf{x}_i) \ . \tag{3.3}$$

Donde $p_\theta(N) \stackrel{\text{def}}{=} \theta^N \exp(-\theta)/N!$ y $\theta = \mathbb{E}_\theta[N] \in \mathbb{R}_+$ es el parámetro de la distribución.

La FIM para este modelo puede ser escrita como:

$$\mathbf{I}_{(\theta,\lambda)} = \begin{pmatrix} \frac{1}{\theta} & 0^T \\ 0 & \theta \mathbf{I}_\lambda \end{pmatrix} \ , \tag{3.4}$$

donde $\mathbf{I}_\lambda$ es la FIM para p_λ. El nuevo mapeo $\hat{\mathcal{G}}_\lambda : \mathcal{X} \times \mathbb{Z}^+ \to \mathbb{R}^{M+1}$ se convierte en:

$$\hat{\mathcal{G}}_\lambda(\mathbf{X}, N) = \frac{1}{\sqrt{\theta}} \begin{pmatrix} N - \theta \\ \sum_{i=1}^N \mathbf{L}_\lambda \nabla_\lambda \log p_\lambda(\mathbf{x}_i) \end{pmatrix} \ . \tag{3.5}$$

De la ecuación (3.5), el significado del término de Poisson se vuelve claro: codifica la desviación de la media del número de elementos en el conjunto. Para problemas con cardinalidad fija, esta formulación se reduce a la codificación FV estándar. En lo que sigue, se pone el foco en el caso $N = 1$ ya que la extensión para conjuntos arbitrarios es sencilla con las definiciones de arriba.

3.5. El modelo mezcla de la familia exponencial

Trabajar con FV requiere la elección apropiada de la forma paramétrica de p_λ y qué modelo elegir depende fuertemente en las particularidades de los datos. En la práctica, a menudo se da el caso de que poco se conoce o se puede asumir acerca de la estructura de los datos más allá del rango de valores para los cuales está definido. Incluso si el modelo está bien formulado, diferentes problemas pueden requerir diferentes niveles de complejidad para poder capturar las sutilezas y particularidades de los datos actuales.

Basándose en estas observaciones, se extiende el modelo de FV considerando distribuciones mezcla de la forma:

$$p_\lambda(x) = \sum_{k=1}^{K} w_k p_k(x), \quad w_k > 0 \; \forall k, \quad \sum_{k=1}^{K} w_k = 1 \; , \tag{3.6}$$

y donde $p_k : \mathcal{X} \to \mathbb{R}_+$ es un miembro de la familia exponencial [Bis06], esto es distribuciones de la forma:

$$p_k(x) \overset{\text{def}}{=} p(x; \boldsymbol{\eta}_k) = h(x) \exp\left[\langle \boldsymbol{\eta}_k, T_k(x) \rangle - \psi(\boldsymbol{\eta}_k)\right] . \tag{3.7}$$

$\boldsymbol{\eta}_k \in \mathbb{R}^q$ es el vector de parámetros naturales, $T_k(x) \in \mathbb{R}^q$ es el vector de estadísticas suficientes para la distribución, $\psi : \mathbb{R}^q \to \mathbb{R}$ es conocida como la función de partición logarítmica y $h : \mathcal{X} \to \mathbb{R}$ es un normalizador. En la tabla 3.1 se muestran algunos ejemplos de distribuciones que son miembros de la familia exponencial.

Siguiendo [KVJ11], como ya se hizo en la sección 2.3 para los FV, se reescriben los pesos de la mezcla como:

$$w_k = \frac{\exp(\alpha_k)}{\sum_{j=1}^{N} \exp(\alpha_j)}$$

para evitar tener que forzar explícitamente la restricción de normalización en la ecuación (3.6).

Para cada elección de $p(\,\cdot\,; \boldsymbol{\eta}_k)$, el vector $\lambda = \left(\alpha_1, \ldots, \alpha_K, \boldsymbol{\eta}_1^T, \ldots, \boldsymbol{\eta}_K^T\right)^T \in \mathbb{R}^{K(q+1)}$ caracteriza completamente la distribución mezcla.

Los parámetros pueden ser fácilmente estimados desde un conjunto finito de muestras usando el algoritmo de EM [RW84] de manera similar a la presentada en la sección 2.3.7. Esto consiste en iteraciones de la forma:

$$\boldsymbol{\eta}_k^{(t+1)} \leftarrow H\left[\frac{\sum_{i=1}^{N} \gamma_k^{(t)}(x_i) T(x_i)}{\sum_{i=1}^{N} \gamma_k^{(t)}(x_i)}\right] \tag{3.8}$$

$$\alpha_k^{(t+1)} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \gamma_k^{(t)}(x_i), \tag{3.9}$$

$$w_k^{(t+1)} \leftarrow \frac{\exp\left(\alpha_k^{(t)}\right)}{\sum_{j=1}^{N} \exp\left(\alpha_j^{(t)}\right)}, \tag{3.10}$$

donde H es un estimador de máxima verosimilitud para $\boldsymbol{\eta}_k$ y $\gamma_k(x)$ es el posterior de la muestra

dada por la k-ésima componente de la mezcla, definido como:

$$\gamma_k(x) \stackrel{\text{def}}{=} \frac{w_k p_k(x)}{\sum_{j=1}^{K} w_j p_j(x)} \tag{3.11}$$

Tabla 3.1: Ejemplos de distribuciones de la familia exponencial unidimensionales (arriba) y multivariadas (abajo). † $\mathcal{S}(D)$ denota el espacio de las matrices SPD de $D \times D$ dimensiones.

Distribución	$\mathcal{X}$	$T(x)$	$\psi(\boldsymbol{\eta})$	$h(x)$	$H(t)$
Gaussiana	$\mathbb{R}$	$\begin{pmatrix} x & x^2 \end{pmatrix}^T$	$-\frac{\boldsymbol{\eta}_1^2}{4\boldsymbol{\eta}_2} - \frac{1}{2}\log(-2\boldsymbol{\eta}_2)$	1	$\begin{pmatrix} \frac{t_1}{t_2 - t_1^2} & -\frac{1}{2}\frac{1}{t_2 - t_1^2} \end{pmatrix}^T$
Bernoulli	$\{0, 1\}$	x	$\log(1 + e^{\boldsymbol{\eta}})$	1	$\log\left(\frac{p}{1-p}\right)$
Exponencial	$\mathbb{R}_+$	x	$-\log(-\boldsymbol{\eta})$	1	$-t$
Poisson	$\mathbb{N}$	x	$e^{\boldsymbol{\eta}}$	$\frac{1}{x!}$	$\log(t)$
Ext. Multivariada	$\mathcal{X}^D$	$\begin{bmatrix} T(x_1) & \dots & T(x_D) \end{bmatrix}^T$	$\sum_{i=1}^D \psi(\boldsymbol{\eta}_i)$	$\prod_{i=1}^D h(x_i)$	$\begin{bmatrix} H(t_1) & \dots & H(t_D) \end{bmatrix}^T$
Dirichlet	$[0, 1]^D$	$\begin{bmatrix} \log(x_1) & \dots & \log(x_D) \end{bmatrix}^T$	$\sum_{i=1}^D \log\Gamma(\boldsymbol{\eta}_i + 1) - \log\Gamma\left[\sum_{i=1}^D (\boldsymbol{\eta}_i + 1)\right]$	1	[Min00]
Wishart†, n dof	$\mathcal{S}(D)$	$\mathbf{x}$	$\log\Gamma_D\left(\frac{n}{2}\right) - \frac{n}{2}\log\lvert\boldsymbol{\eta}\rvert$	$\lvert\mathbf{x}\rvert^{(n-D-1)/2}$	$-\frac{n}{2}t^{-1}$

3.5.1. Extensión multivariada

Para datos multivariados (vectoriales), se pueden extender las distribuciones unidimensionales a un número arbitrario de dimensiones de la siguiente forma. Sea $p_i : \mathcal{X} \to \mathbb{R}_+$ una distribución (univariada) miembro de la familia exponencial con vector parámetros $\boldsymbol{\eta}_i$ como se definió en la ecuación (3.7); suponiendo que no hay correlación entre las dimensiones, se puede definir su extensión D-dimensional $\tilde{p}_k : \mathcal{X}^D \to \mathbb{R}_+$ como:

$$\tilde{p}_k(\mathbf{x}) = \prod_{i=1}^{D} p_i(x_i). \tag{3.12}$$

Lo primero que se debe verificar, es que $\tilde{p}_k$ también pertenece a la familia exponencial. Sea $\mathbf{x} = [x_1, \ldots, x_D]^T$ y $\boldsymbol{\eta} = [\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_D]^T$, se tiene que:

$$\begin{aligned}
\tilde{p}_k(\mathbf{x}) &= \prod_{i=1}^{D} p_i(x_i) = \prod_{i=1}^{D} h_i(x_i) \exp\left[\langle \boldsymbol{\eta}_i, T_i(x_i)\rangle - \psi_i(\boldsymbol{\eta}_i)\right] \\
&= \left[\prod_{i=1}^{D} h(x_i)\right] \left[\prod_{i=1}^{D} \exp\left[\langle \boldsymbol{\eta}_i, T_i(x_i)\rangle - \psi_i(\boldsymbol{\eta}_i)\right]\right] \\
&= \left[\prod_{i=1}^{D} h(x_i)\right] \exp\left[\sum_{i=1}^{D} \langle \boldsymbol{\eta}_i, T_i(x_i)\rangle - \sum_{i=1}^{D} \psi_i(\boldsymbol{\eta}_i)\right].
\end{aligned}$$

Definiendo $h(\mathbf{x}) = \prod_{i=1}^{D} h_i(x_i)$, $T(\mathbf{x}) = [T_1(x_1), \ldots, T_D(x_D)]^T$ y $\psi(\boldsymbol{\eta}) = \sum_{i=1}^{D} \psi_i(\boldsymbol{\eta}_i)$, $\tilde{p}_k(\mathbf{x})$ toma la forma de la ecuación (3.7):

$$\tilde{p}_k(\mathbf{x}) = h(\mathbf{x}) \exp\left[\langle \boldsymbol{\eta}, T(\mathbf{x})\rangle - \psi(\boldsymbol{\eta})\right],$$

por lo tanto también pertenece a la familia exponencial. De ahora en adelante solo se usarán distribuciones de la familia exponencial multivariadas las cuales se denotarán con p_k y con p_λ se denotarán mezclas de K distribuciones.

Si se consideran mezclas de pdfs definidas como en la ecuación (3.12), la esperanza se puede expresar de la siguiente manera:

$$\begin{aligned}
\mathbb{E}_{\mathbf{x}\sim p_\lambda}[\mathbf{x}] &= \int_{\mathcal{X}^D} p_\lambda(\mathbf{x})\ \mathbf{x}\ d\mathbf{x} = \int_{\mathcal{X}^D} \left[\sum_{k=1}^{K} w_k p_k(\mathbf{x})\right] \mathbf{x}\ d\mathbf{x} \\
&= \sum_{k=1}^{K} w_k \int_{\mathcal{X}^D} p_k(\mathbf{x})\ \mathbf{x}\ d\mathbf{x} = \sum_{k=1}^{K} w_k \mathbb{E}_{\mathbf{x}\sim p_k}[\mathbf{x}] \\
\mathbb{E}_{\mathbf{x}\sim p_\lambda}[\mathbf{x}] &= \sum_{k=1}^{K} w_k \boldsymbol{\mu}_k,
\end{aligned}$$

donde $\boldsymbol{\mu}_k = \mathbb{E}_{\mathbf{x}\sim p_k}[\mathbf{x}]$; y la covarianza se puede expresar como:

$$\begin{aligned}
\text{cov}_{\mathbf{x}\sim p_\lambda}[\mathbf{x}] &= \mathbb{E}_{\mathbf{x}\sim p_\lambda}\left[\mathbf{x}\mathbf{x}^T\right] - \mathbb{E}_{\mathbf{x}\sim p_\lambda}[\mathbf{x}]\mathbb{E}^T_{\mathbf{x}\sim p_\lambda}[\mathbf{x}] \\
&= \int_{\mathcal{X}^D} p_\lambda(\mathbf{x})\ \mathbf{x}^2\ d\mathbf{x} - \mathbb{E}_{\mathbf{x}\sim p_\lambda}[\mathbf{x}]\mathbb{E}^T_{\mathbf{x}\sim p_\lambda}[\mathbf{x}]
\end{aligned}$$

$$
\begin{aligned}
&= \int_{\mathcal{X}^D} \sum_{k=1}^{K} w_k p_k(\mathbf{x})\ \mathbf{x}^2\ d\mathbf{x} - \mathbb{E}_{\mathbf{x}\sim p_\lambda}[\mathbf{x}]\mathbb{E}^T_{\mathbf{x}\sim p_\lambda}[\mathbf{x}] \\
&= \sum_{k=1}^{K} w_k \int_{\mathcal{X}^D} p_k(\mathbf{x})\ \mathbf{x}^2\ d\mathbf{x} - \mathbb{E}_{\mathbf{x}\sim p_\lambda}[\mathbf{x}]\mathbb{E}^T_{\mathbf{x}\sim p_\lambda}[\mathbf{x}] \\
&= \sum_{k=1}^{K} w_k \mathbb{E}_{\mathbf{x}\sim p_k}[\mathbf{x}^2] - \mathbb{E}_{\mathbf{x}\sim p_\lambda}[\mathbf{x}]\mathbb{E}^T_{\mathbf{x}\sim p_\lambda}[\mathbf{x}]
\end{aligned}
$$

$$
\text{cov}[\mathbf{x}] = \sum_{k=1}^{K} w_k\big(\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k\boldsymbol{\mu}_k^T\big) - \mathbb{E}_{\mathbf{x}\sim p_\lambda}[\mathbf{x}]\mathbb{E}_{\mathbf{x}\sim p_\lambda}[\mathbf{x}]^T \tag{3.13}
$$

donde $\boldsymbol{\Sigma}_k = \text{cov}_{\mathbf{x}\sim p_k}[\mathbf{x}]$. Ya que la covarianza en (3.13) no es diagonal, mezclas de distribuciones como las de la ecuación (3.12) pueden aún capturar en parte las correlaciones entre las dimensiones [Bis06] mitigando la fuerte suposición realizada en un principio.

3.6. Vectores de Fisher de la familia exponencial

Para derivar el mapeo FV para el modelo de la ecuación (3.6) se necesita computar el gradiente con respecto a λ y derivar una expresión para el normalizador de la ecuación (3.1). Para el gradiente se tiene que (ver apéndice D):

$$
\partial_{\alpha_k} \log p_\lambda(\mathbf{x}) = \gamma_k(\mathbf{x}) - w_k \tag{3.14}
$$

$$
\nabla_{\boldsymbol{\eta}_k} \log p_\lambda(\mathbf{x}) = \gamma_k(\mathbf{x})[T(\mathbf{x}) - \nabla_{\boldsymbol{\eta}_k}\psi(\boldsymbol{\eta}_k)]. \tag{3.15}
$$

El cómputo de $\mathbf{L}_\lambda$ es bastante costoso ya que requiere el cómputo de la descomposición de Cholesky y la inversión de una matriz de dimensionalidad $K(q+1) \times K(q+1)$. A medida que aumenta el número de componentes de la mezcla o la dimensionalidad de la entrada, esto se vuelve rápidamente impráctico.

Para hacer el problema tratable, siguiendo [PD07] y asumiendo que la asignación de muestras a las componentes del vocabulario es casi "dura" se plantea una aproximación diagonal por bloques a la FIM $\mathbf{L}_\lambda$.

Suponiendo una aproximación de asignamiento duro de muestras a componentes de la mezcla:

$$
\gamma_i(\mathbf{x})\gamma_j(\mathbf{x}) \approx \begin{cases} \gamma_i(\mathbf{x}) & \text{si } i = j \\ 0 & \text{en otro caso.} \end{cases} \tag{3.16}
$$

Definiendo la siguiente igualdad para ayudar al entendimiento:

$$
\ell_\xi \overset{\text{def}}{=} \nabla_\xi \log\ p_\lambda(\mathbf{x})
$$

Para mezclas de distribuciones como en la ecuación (3.6), la FIM está compuesta por los siguientes bloques: *a)* $\mathbb{E}[\ell_\alpha\ell_\alpha]$, *b)* $\mathbb{E}[\ell_\alpha\ell_{\boldsymbol{\eta}_i}]$ y *c)* $\mathbb{E}\big[\ell_{\boldsymbol{\eta}_i}\ell_{\boldsymbol{\eta}_j}\big]$, para $i, j = 1,\ \ldots,\ K$. El desarrollo de los casos *a* y b se puede ver en [SPMV13].

Para el caso *c*, todas las entradas $i \neq j$ son ceros bajo la suposición de asignamiento duro. Para $i = j$, se tiene:

$$
\mathbb{E}\big[\ell_{\boldsymbol{\eta}_i}\ell_{\boldsymbol{\eta}_i}{}^T\big] = \mathbb{E}\Big[[\nabla_{\boldsymbol{\eta}_i} \log\ p_\lambda(\mathbf{x})][\nabla_{\boldsymbol{\eta}_i} \log\ p_\lambda(\mathbf{x})]^T\Big]
$$

y usando la ecuación (3.15):

$$\begin{aligned}\mathbb{E}\big[\ell_{\boldsymbol{\eta}_i}{\ell_{\boldsymbol{\eta}_i}}^T\big] &= \mathbb{E}\Big[\gamma_i(\mathbf{x})\tilde{T}_i(\mathbf{x})\gamma_i(\mathbf{x})\tilde{T}_i^T(\mathbf{x})\Big] \\ &= \int_{\mathcal{X}} \gamma_i(\mathbf{x})\gamma_i(\mathbf{x})\tilde{T}_i(\mathbf{x})\tilde{T}_i^T(\mathbf{x})p_\lambda(\mathbf{x})d\mathbf{x}\end{aligned}$$

y usando la aproximación de asignamiento duro de la ecuación (3.16):

$$\mathbb{E}\big[\ell_{\boldsymbol{\eta}_i}{\ell_{\boldsymbol{\eta}_i}}^T\big] \approx \int_{\mathcal{X}} \gamma_i(\mathbf{x})\tilde{T}_i(\mathbf{x})\tilde{T}_i^T(\mathbf{x})p_\lambda(\mathbf{x})d\mathbf{x} \tag{3.17}$$

$$\approx \int_{\mathcal{X}} \frac{w_i p_i(\mathbf{x})}{p_\lambda(\mathbf{x})}\tilde{T}_i(\mathbf{x})\tilde{T}_i^T(\mathbf{x})p_\lambda(\mathbf{x})d\mathbf{x} \tag{3.18}$$

$$\approx w_i \int_{\mathcal{X}} p_i(\mathbf{x})\tilde{T}_i(\mathbf{x})\tilde{T}_i^T(\mathbf{x})d\mathbf{x} = w_i\mathbf{I}_i \tag{3.19}$$

donde $\tilde{T}_i(\mathbf{x}) \stackrel{\text{def}}{=} T(\mathbf{x}) - \nabla_{\eta_i}\psi(\boldsymbol{\eta}_i)$ y $\mathbf{I}_i$ es la FIM correspondiente a la componente i-ésima considerada en forma aislada.

Usando esto $\mathbf{L}_\lambda$ puede ser expresada como:

$$\mathbf{L}_\lambda \approx \begin{pmatrix} \mathbf{L}_\alpha & 0 & \cdots & 0 \\ 0 & \mathbf{L}_1/\sqrt{w_1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{L}_K/\sqrt{w_K} \end{pmatrix}, \tag{3.20}$$

donde $\mathbf{L}_\alpha \stackrel{\text{def}}{=} \operatorname{diag}\Big(\frac{1}{\sqrt{w_1}}, \ldots, \frac{1}{\sqrt{w_K}}\Big)$ y $\mathbf{L}_i$ es la matriz que resulta de $\mathbf{I}_i^{-1} = \mathbf{L}_i^T\mathbf{L}_i$, con $\mathbf{I}_i$ la FIM para la i-ésima componente de la mezcla considerada en forma independiente.

Estos $\mathbf{L}_i$ puede ser rápidamente estimados tomando muestras de cada componente separadamente y computando las covarianzas de los "scores" de las muestras, lo cual es más eficiente que tomar muestras de las distribuciones de las mezclas y aproximar los normalizadores en bloques directamente. El cómputo del normalizador (3.20) requiere la inversión de K matrices SPD de dimensión $q \times q$ en vez de la matriz completa.

Usando las ecuaciones (3.14)–(3.20), se define una familia de modelos de la forma:

$$\mathcal{G}_\lambda(\mathbf{x}) \stackrel{\text{def}}{=} \big[g_\alpha(\mathbf{x})^T, g_1(\mathbf{x})^T, \ldots, g_K(\mathbf{x})^T\big]^T,$$

donde:

$$g_\alpha(\mathbf{x}) = \left[\frac{\gamma_1(\mathbf{x}) - w_1}{\sqrt{w_1}}, \ldots, \frac{\gamma_K(\mathbf{x}) - w_K}{\sqrt{w_K}}\right]^T, \tag{3.21}$$

$$g_i(\mathbf{x}) = \frac{\gamma_i(\mathbf{x})}{\sqrt{w_i}}\mathbf{L}_i[T(\mathbf{x}) - \nabla_i\psi(\eta_i)], \quad i = 1, \ldots, K\,, \tag{3.22}$$

y se nombra a esta familia *vectores de Fisher de la familia exponencial*.

Una interpretación de los eFV Para distribuciones que pertenecen a la familia exponencial se cumple que $\nabla_{\boldsymbol{\eta}}\psi(\boldsymbol{\eta}) = \mathbb{E}_{\mathbf{x}\sim p}[T(\mathbf{x})]$ (ver demostración en el apéndice E). Los gradientes en la ecuación (3.15) pueden entonces ser escritos como:

$$\nabla_{\boldsymbol{\eta}_k} \log p_\lambda(\mathbf{x}) = \gamma_k(\mathbf{x})[T(\mathbf{x}) - \mathbb{E}_{\mathbf{x}\sim p_k}[T(\mathbf{x})]]\,. \tag{3.23}$$

Desde lo cual se sigue que el eFV codifica la desviación con respecto al valor esperado de la estadística suficiente de la muestra con respecto al modelo p_λ.

3.6.1. Clasificación lineal y espacios de entrada finitos

Para cerrar esta sección, se discute el problema de clasificación lineal con eFV cuando el espacio de entrada $\mathcal{X}$ es finito, por ejemplo si existe un $n \in \mathbb{N}$ tal que $\imath : \mathcal{X} \to \{0, 1, \cdots, n-1\}$ es una biyección. Un clasificador lineal f_w actuando sobre la codificación eFV de un conjunto $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ predecirá un "score":

$$f_w[\mathcal{G}_\lambda(\mathbf{X})] = w_0 + w^T \mathcal{G}_\lambda(\mathbf{X}) = w_0 + \sum_{i=1}^{N} \left[w^T \mathcal{G}_\lambda(\mathbf{x}_i)\right] \tag{3.24}$$

Ya que $\mathcal{X}$ es finito, se pueden precomputar los productos punto para cada $\mathbf{x}_i \in \mathcal{X}$ y construir una tabla (indexada por $\imath(\mathbf{x})$) tal que, a tiempo de evaluación, estos se reduzcan a una simple búsqueda en una tabla. En este caso, el costo de clasificar una nueva muestra es independiente del número de componentes K de la mezcla. Esto va un paso más allá que las propuestas realizadas en [CVS13, OVS14], ya que incluso ni se necesita computar el eFV para los elementos del conjunto.

El mismo enfoque puede ser aplicado para indexar las normas de los eFV asociados con cada $\mathbf{x}_i \in \mathcal{X}$ para poder computar una aproximación a la $\mathbf{L}_p$-normalización como en [OVS14].

3.7. Experimentos

En esta sección se evalúan diferentes aspectos de la codificación eFV así como también su comportamiento global sobre dos problemas de clasificación desafiantes. Por el momento solo se muestran experimentos de clasificación de imágenes genéricos, para demostrar la efectividad y robustez del método propuesto dejando las aplicaciones a problemas de AP para los siguientes capítulos de esta tesis.

Se corrieron experimentos usando diferentes combinaciones de características locales y distribuciones mezcla para mostrar la flexibilidad y generalidad del enfoque. Primero se describen los conjuntos de datos utilizados en las evaluaciones y luego la configuración experimental. Por último se reportan los resultados cuantitativos.

3.7.1. Conjuntos de datos

El primer conjunto de datos, llamado Pascal VOC 2007 [EZW+07] contiene alrededor de 10K imágenes representando 20 categorías diferentes de objetos divididas en 3 conjuntos, entrenamiento (2501 imágenes), validación (2510 imágenes) y evaluación (4952 imágenes). En la figura 3.1 se muestran ejemplos de las imágenes pertenecientes a este conjunto de datos. A pesar de su relativo bajo tamaño, este conjunto se mantiene como uno de los más desafiantes en la literatura [TE11]. Para la evaluación, se siguen los procedimientos recomendados en [EZW+07] los cuales consisten en ajustar los parámetros usando los conjuntos de entrenamiento/validación y reportar los resultados sobre el conjunto de evaluación. La precisión en la clasificación es medida usando una métrica conocida como media de la precisión promedio (mAP, por su denominación en inglés) la cual es computada con los "scripts" que se adjuntan con los datos.

El segundo conjunto de datos, conocido como KTH-TIPS2-a [CHM05] contiene 4395 imágenes de 11 materiales con texturas diferentes: *papel aluminio, pan negro, corderoy, corcho, algodón, galletas, hoja de lechuga, lino, pan blanco, madera y lana*, adquiridas bajo diferentes

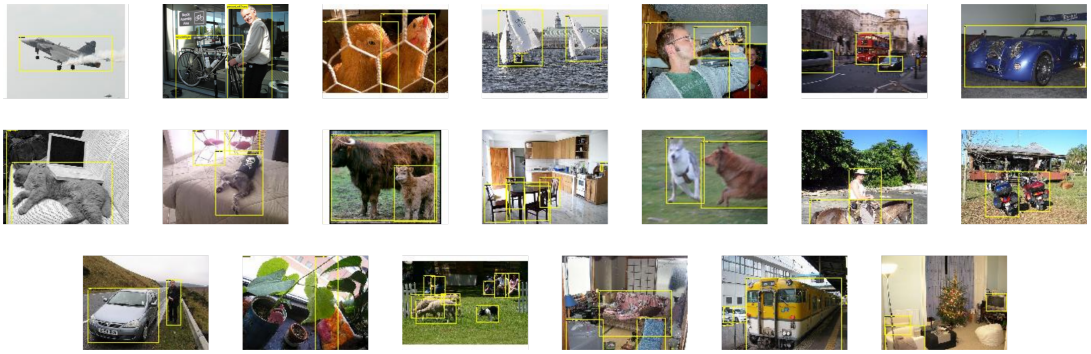


Figura 3.1: Ejemplos de imágenes de cada una de las 20 clases del conjunto de datos Pascal VOC 2007. Los recuadros amarillos indican cuales son los objetos de interés en cada imagen y como se puede apreciar, en algunas de ellas hay más de un objeto de interés.

escalas, poses y condiciones de iluminación. Las imágenes son divididas en 4 subconjuntos (muestras *a, b, c* y *d*). En la figura 3.2 se muestran algunos ejemplos de las texturas que componen el conjunto. Para la evaluación se usa el protocolo estándar, propuesto en [CHM05], que consiste en tomar cada vez una de las muestras para evaluación y las 3 restantes para entrenamiento. Para cada corrida, se realiza un ajuste de parámetros sobre los 3 subconjuntos de entrenamiento usando validación cruzada con 5 iteraciones. Una vez ajustados los parámetros, el desempeño se evalúa como la exactitud promedio sobre las 4 corridas. También se reporta la exactitud para cada corrida individual.

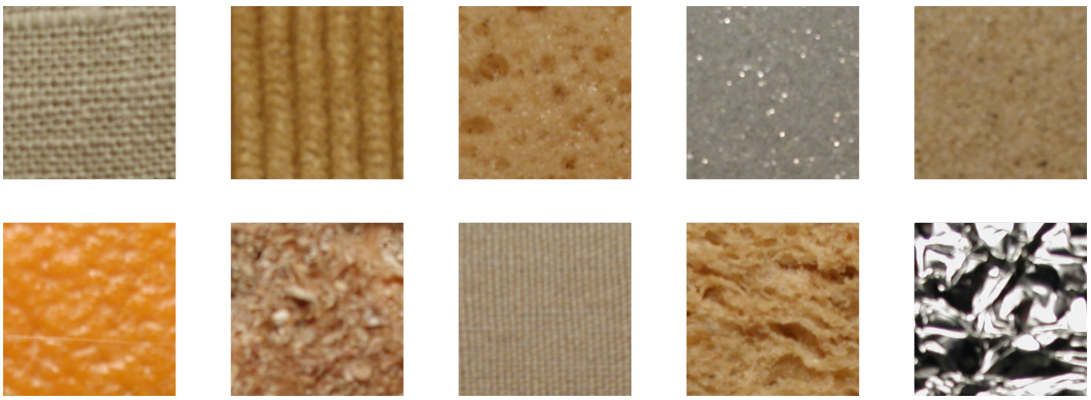


Figura 3.2: Ejemplos de imágenes de texturas del conjunto de datos KTH-TIPS2-a.

3.7.2. Configuración experimental

En esta sección se describe el esquema general utilizado, pero debido a que se usaron diferentes combinaciones de características locales y distribuciones mezcla, los detalles de cada elección particular son dados en la subsección correspondiente. En la figura 3.3 se muestra un diagrama de flujo de los distintos pasos para realizar la codificación eFV. En la misma, la línea de trazos indica que el algoritmo de PCA es opcional para algunos de los descriptores.

- *Características locales*. Dada una imagen, se computa una pirámide de resolución como se muestra en la figura 2.3 con 5 niveles y un factor de disminución de escala de $2^{-1/2}$. Para

cada capa, se extraen descriptores locales desde parches de 24×24 pixeles muestreados en forma regular usando un paso de 6 pixeles de forma similar a como se muestra en la figura 2.1.

- *Modelo mezcla*. Para ajustar los parámetros se usa el algoritmo de EM sobre un conjunto de 1M de muestras aleatorias tomadas del conjunto de entrenamiento. Para inicializar las iteraciones de EM, se corrió K-Means sobre un subconjunto de los datos y se usaron la proporción y las estadísticas suficientes de las muestras asignadas a cada cluster como estimación inicial para los w_is y η_is respectivamente.

- *Codificación eFV*. Se computa el normalizador para los eFV muestreando de cada componente como se describió en la sección 3.6. El parámetro de Poisson θ se fija como el número promedio de muestras extraídas de las imágenes del conjunto de entrenamiento. Siguiendo [PSM10], se aplica al vector resultante la transformación de raíz cuadrada con signo y la normalización L_2 como ya se explicó en la sección 2.3.8 del capítulo anterior.

- *Clasificadores*. Se utilizaron SVM lineales implementados en la librería LIBLINEAR [FCH+08] en una estrategia de *uno-contra-todos*.

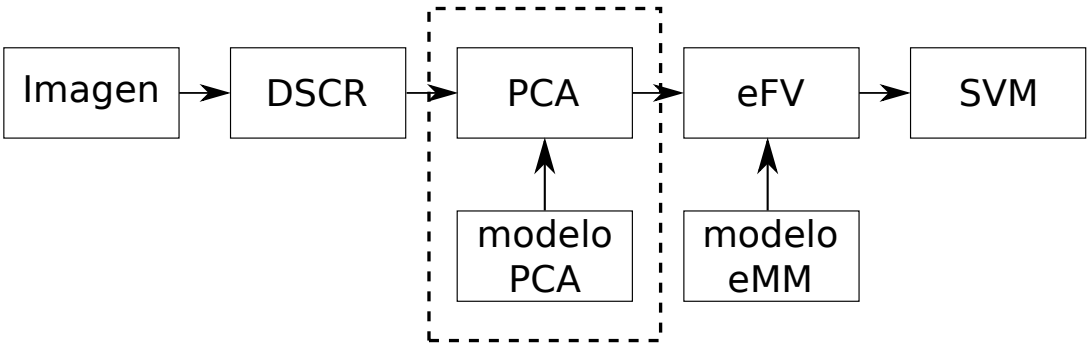


Figura 3.3: Diagrama de flujo de la codificación eFV utilizada en los experimentos. La línea de trazos indica que el algoritmo de PCA es opcional.

A continuación, se usa eFV-X para denotar el eFV derivado de una *X*MM, por ejemplo eFV-G denotará la codificación eFV computada desde una GMM y eFV-B desde una mezcla de distribuciones de Bernoulli (BMM).

3.7.3. Efecto de la cardinalidad de la muestra

Primero se evalúa el efecto de incluir el término de Poisson en la formulación del modelo eFV, ecuación (3.5). Ya que el gradiente con respecto a θ agrega una sola dimensión extra al, en general de alta dimensionalidad, eFV, no se pueden esperar grandes mejoras en comparación con la formulación básica. A pesar de esto, esta dimensión extra puede agregar información valiosa relacionada con el tamaño de los objetos. La idea es que, ya que los parches de las imágenes están muestreados regularmente (ver figura 2.1), la cardinalidad de la muestra se relaciona con el tamaño del objeto en la imagen. Para validar esta hipótesis se corrieron experimentos en Pascal VOC 2007, usando las cajas limitantes provistas en este conjunto de datos para enfocar el cómputo sobre los parches del "primer plano" solamente. Se dice que los parches de una imagen pertenecen al primer plano si caen adentro de cualquiera de las cajas limitantes provistas para esa imagen.

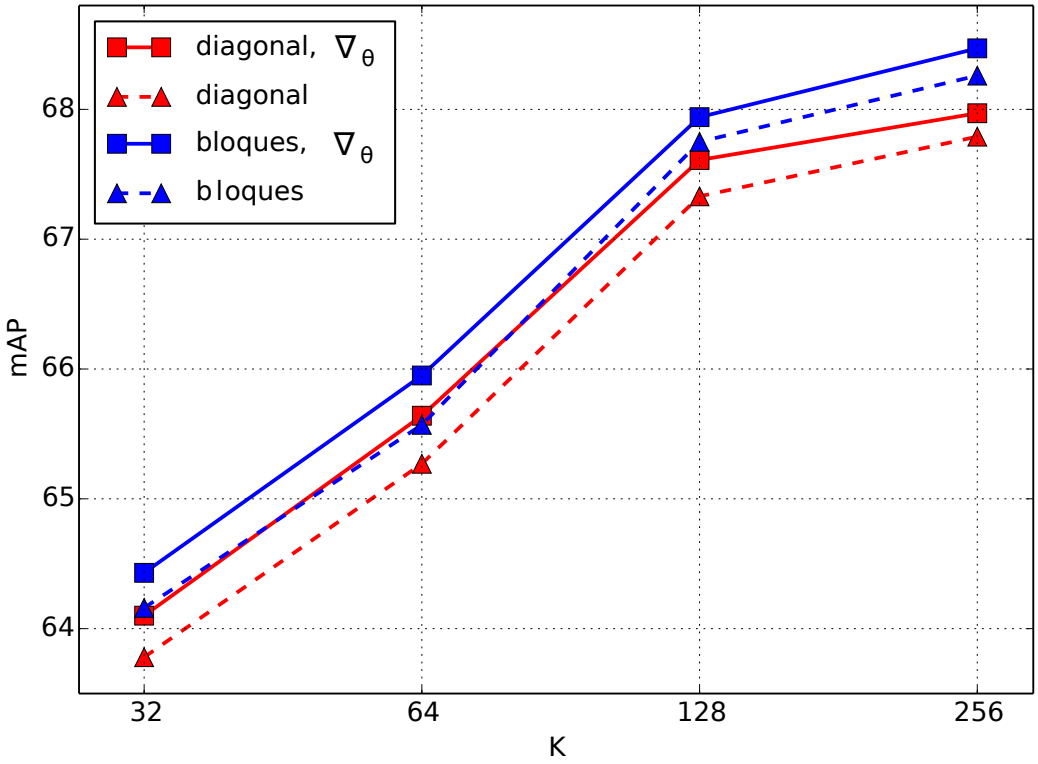


Figura 3.4: Efecto de las diferentes formulaciones de eFV medidas en el conjunto de datos Pascal VOC 2007 computando solo descriptores sobre las cajas limitantes. Normalizadores diagonal (rojo) y diagonal por bloques (azul); con (sólido) y sin (rayas) el gradiente con respecto a θ.

Para este experimento se usan descriptores SIFT (4×4 celdas de 6×6 pixeles cada una) proyectados a 64 dimensiones con PCA (SIFT-PCA, $\mathcal{X} = \mathbb{R}^{64}$) y codificación eFV-G [3]. Además del normalizador diagonal por bloques de la ecuación (3.20), se considera el modelo diagonal completo obtenido restringiendo a que cada bloque también sea diagonal. En la figura 3.4 se muestran los resultados del incremento de los valores de K.

Como se puede observar, hay un pequeño pero notable efecto con el agregado del término extra de Poisson a la representación. Notablemente, este término extra representa solo una pequeña fracción de la dimensionalidad de la codificación ($0{,}003\,\%$ para $K = 256$). Cuando se corren los experimentos usando el conjunto completo de descriptores no se observa ninguna mejora significativa, como se esperaba. En este caso, la cardinalidad de la muestra no lleva ninguna información discriminativa. Respecto a las diferentes normalizaciones, se puede ver que el uso de la formulación diagonal por bloques conduce a resultados levemente mejores ($+0{,}3$ puntos absolutos).

En lo que sigue, se usa por defecto el modelo de normalización diagonal por bloques para eFV-G.

[3] Notar que la extensión multivariada de gaussianas de 1 dimensión (Sección 3.5.1) conduce a una mezcla con covarianzas diagonales, como en el FV estándar.

3.7.4. Clasificación con características binarias

El foco de esta sección es el problema de clasificación usando características binarias. Para ello se reportan resultados sobre ambos conjunto de datos, PASCAL VOC 2007 y KTH-TIPS2-a. Para la codificación se usó una mezcla de pdfs de Bernoulli multivariadas [4]. Para la inicialización del ajuste del modelo, se utilizó el método de prototipo aleatorio de [JGHV04] con un factor de mezcla de $0{,}5$. Primero, se muestran resultados en Pascal VOC 2007 usando características binarias y binarizadas. Luego se evalúa el uso de descriptores binarios para clasificación de texturas en el desafiante conjunto de datos KTH-TIPS2-a.

Pascal VOC 2007 En la figura 3.5 se muestran los resultados sobre PASCAL VOC 2007 para las siguientes configuraciones: un sistema base basado en SIFT-PCA y codificación eFV-G; dos sistemas basados en características SIFT-PCA binarizadas [5]: el primero basado en modelar las características locales con BMMs (SIFT-PCA-bin, eFV-B), y el segundo tratando a las datos binarios como valores reales y usando GMM como modelo (SIFT-PCA-bin, eFV-G). Este último puede ser visto como una heurística para la clasificación con características binarias. Finalmente, también se reporta resultados usando Características Elementales Independientes Robustas Binarias (BRIEF, por su denominación en inglés) de 256 bits [CLO$^+$12] y eFV-B (BRIEF-256, eFV-B).

Con el sistema base se logra un desempeño de $59{,}5\,\%$ mAP para $K = 512$. Esto es comparable con el mejor resultado publicado para este conjunto de datos usando características SIFT-PCA y FV gaussianos [SPMV13]. Si se consideran los dos sistemas basados en características SIFT-PCA-bin, se observa que para el sistema que usa GMM el rendimiento alcanza un pico de $47{,}5$ en $K = 128$ mientras que para el sistema basado en BMMs hay un aumento constante, alcanzando $54{,}8\,\%$ de mAP en $K = 512$. Notar que, para el mismo valor de K, eFV-B tiene aproximadamente la mitad de la dimensionalidad que eFV-G (ya que los parámetros de la GMM incluyen la media y las varianzas en forma explícita). Para la misma dimensionalidad, el desempeño logrado por eFV-B ($K = 512$) esta solo 5 puntos absolutos por debajo de la del sistema base. Desde el punto de vista del almacenamiento, la memoria requerida por las características binarizadas es 32 veces menor (para flotantes de precisión simple) que para su contraparte real.

Para BRIEF, se logró un máximo rendimiento de $39{,}4\,\%$ mAP ($K = 256$). Este resultado supera al mejor publicado en la literatura en este conjunto de datos usando este tipo de descriptores. Por ejemplo, [CAGA14] reporta una mAP de $36{,}2\,\%$ con el modelo presentado en [ATC$^+$13] basado en un vocabulario de 1024 palabras, pirámides de resolución y SVMs no lineales.

Para eFV-B no se observa ninguna diferencia significante entre los normalizadores diagonales y diagonales por bloque.

KTH-TIPS2-a Los Patrones Locales Binarios (LBP, por su denominación en inglés) [OPM02] son descriptores binarios populares para clasificación de texturas. En este experimento se usó LBP con parámetros de configuración $R = 1$ y $P = 8$ computados densamente y en múltiples escalas de resolución (se usó la misma configuración de pirámides que antes). Se reportan resultados de un sistema base usando Histogramas de Patrones Locales Binarios (LBPH, por su denominación en inglés) y para uno basado en eFV-B ($K = 128$). Para LBPH se usan las mismas normalizaciones que para eFV-B ya que se observa que mejora el rendimiento. Los resultados se muestran en las dos primeras filas de la tabla 3.2. Considerado separadamente, el sistema

[4] La extensión multivariada de la distribución de Bernoulli es también conocida en la literatura como distribución *multinoulli* [Mur12].

[5] Obtenidas aplicando la función $b(z) \stackrel{\text{def}}{=} \max\{0, \operatorname{sign}(z)\}$ dimensión a dimensión.

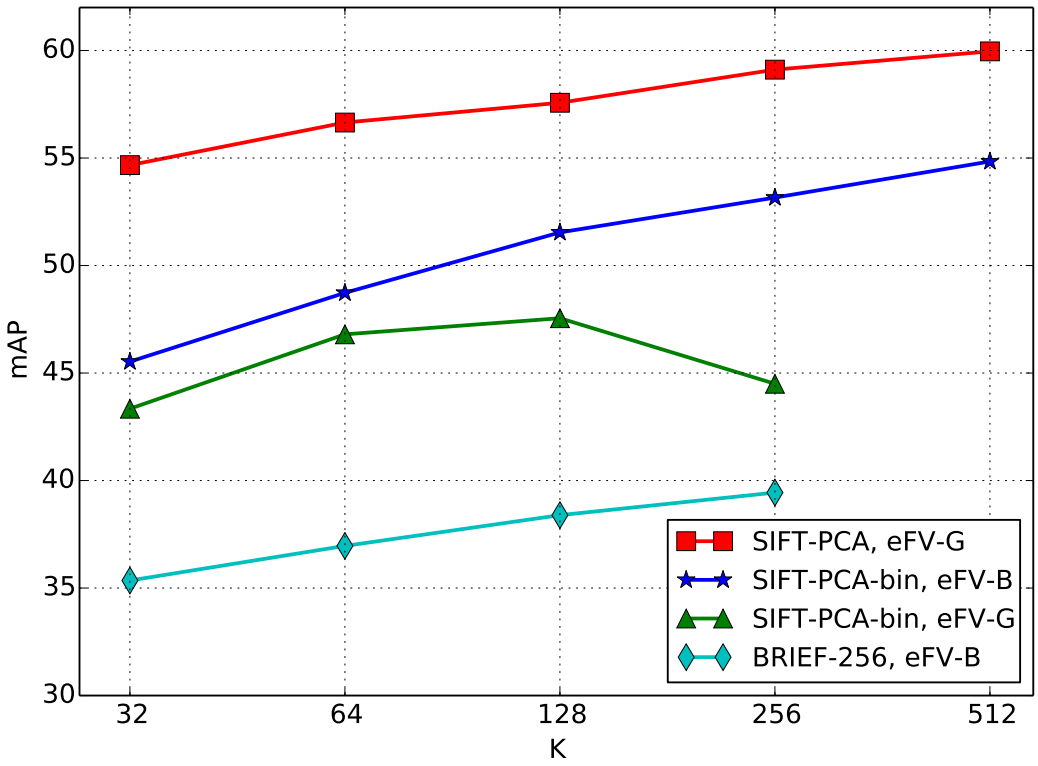


Figura 3.5: eFV sobre descriptores binarios evaluados en PASCAL VOC 2007. Sistema base SIFT-PCA FV (rojo); eFV-B (azul) y eFV-G (verde) sobre vectores SIFT-PCA binarizados. Características BRIEF y codificación eFV-B (cian).

basado en eFV-B obtiene una mejora de alrededor 2 puntos en tres de las 4 muestras y una leve mejora en el promedio de todas las corridas.

3.7.5. Descriptores del tipo matrices simétricas positiva definidas

Para este experimento se considera una variación de los Descriptores de Covarianza (DCOV) propuestos en [TPM06] y se los codifica con eFV basados en mezclas de distribuciones de Wishart. Para cada parche, se computa la covarianza de la muestra de las siguientes características de los pixeles:

$$F(x,y) = \begin{pmatrix} x & y & \sigma & I & |I_x| & |I_y| & |I_{xx}| & |I_{xy}| & |I_{yy}| \end{pmatrix}. \tag{3.25}$$

Aquí, (x,y) y σ son las coordenadas del pixel y la escala del parche respectivamente; $I \stackrel{\text{def}}{=} I(x,y)$ es la intensidad de la imagen en (x,y), $I_\xi \stackrel{\text{def}}{=} \frac{\partial I}{\partial \xi}$ y $I_{\xi\zeta} \stackrel{\text{def}}{=} \frac{\partial^2 I}{\partial \xi \partial \zeta}$ denotan la primera y segunda derivada respectivamente. Esto resulta en un descriptor por parche el cual es una matriz SPD de dimensionalidad 9×9.

En la tabla 3.2 se muestra el resultado obtenido usando descriptores DCOV y eFV-W ($K = 64$) para normalizadores diagonales y diagonales por bloque. El parámetro n, que representa los grados de libertad de las pdf de Wishart (ver tabla 3.1), fue puesto a 576, esto es igual al número de pixeles dentro del parche. Se reportan resultados para un sistema base usando características SIFT-PCA y eFV-G ($K = 256$). Como comparación, también se muestran algunos

Tabla 3.2: Resultados de clasificación sobre KTH-TIPS2-a (ver el texto para los detalles).

Método	a	b	c	d	Exactitud
LBPH	71.6	70.4	79.0	63.6	71.1 (5.5)
LBP, eFV-B	73.2	72.7	76.6	65.0	71.9 (4.2)
SIFT-PCA, eFV-G	81.5	78.6	76.3	73.7	77.5 (2.9)
DCOV, eFV-W (diag)	85.5	76.2	74.2	69.7	76.4 (5.6)
DCOV, eFV-W (block)	85.9	71.2	73.2	73.9	77.5 (5.1)
LHS [SuHJ12]	–	–	–	–	73.0 (4.7)
DeCAF [CMK$^+$14]	–	–	–	–	78.4 (2.0)

resultados reportados recientemente en la literatura: el modelo LHS de [SuHJ12] y otro basado en características obtenidas de redes profundas [CMK$^+$14]. El primero está basado en el llamado "vectores diferencia" y FV mientras que el segundo toma como características la salida de una red profunda (DeCAF) de la cual ha sido removida la última capa totalmente conectada (ver capítulo 4 para una introducción a las redes profundas). En [CMK$^+$14], los resultados también se reportan para un sistema basado en el cómputo de SIFT denso y FV ($82{,}2$ $(4{,}6)$) y para la combinación de DeCAF y FV ($84{,}7$ $(1{,}5)$). Para el método basado en eFV, solo se usan características computadas en un solo canal para realizar una comparación justa.

De la tabla, se vé que para eFV-W, con la normalización diagonal por bloques se obtiene el mejor rendimiento. En este caso, el sistema basado en DCOV y eFV-W logra un desempeño que está a la par con el estado del arte para este conjunto de datos.

3.7.6. Histogramas locales

Ahora se aplica la codificación al caso de características locales en el símplex estándar de $d-1$ dimensiones y mezclas de pdfs de Dirichlet (DMM). Aquí, se consideran descriptores de histogramas de colores (ColH) computados de la siguiente manera: la imagen es dividida en parches y estos a su vez son divididos en celdas de 4×4. De cada celda, se computa un histograma RGB normalizado usando 4 bins por canal de color. Finalmente, las características a nivel de celda son concatenadas y renormalizadas para que tengan norma L_1 unitaria. Los descriptores resultantes tienen 192 dimensiones.

Solo se consideran codificaciones del tipo eFV-G y eFV-D y se fija el número de componentes a $K = 512$. Sobre PASCAL VOC 2007, el sistema basado en eFV-G alcanzó una mAP de $39{,}7\,\%$ mientras que el basado en eFV-D obtuvo una mAP de $41{,}0\,\%$. Valores similares de ganancia fueron observados para valores de K que van de 32 a 512 componentes incluso cuando, para el mismo K la dimensionalidad de los eFV-G es al menos el doble que la dimensionalidad de los eFV-D.

3.8. Conclusiones

En este capítulo se propuso un formalismo para la codificación de imágenes que extiende FV a mezclas de pdfs no gaussianas. Este modelo provee una estructura unificada para la representación de imágenes usando características locales definidas sobre dominios de entrada generales. El modelo fue evaluado empíricamente sobre dos conjuntos de datos desafiantes para modelos basados en mezclas de pdfs gaussianas, Bernoulli, Wishart y Dirichlet. Los resultados

muestran la gran flexibilidad y el poder de modelado del enfoque. Este modelo será utilizado en los siguientes capítulos para resolver diferentes problemas de AP.

Capítulo 4

Redes neuronales convolucionales

Índice

4.1. Introducción

Las CNN son variantes del perceptrón multicapa inspiradas biológicamente, las cuales tratan de imitar la corteza visual animal. Se sabe por estudios que la corteza visual contiene un arreglo complejo de células, las cuales son sensitivas a pequeñas regiones del campo visual. Grupos de estas células son llamadas campos receptivos [HW68] y son emulados en las CNN con filtros de convolución.

Estos tipos de redes son conocidas desde hace mucho tiempo, aunque su uso en aplicaciones complejas se ha visto relegado en el tiempo debido a que este tipo de modelos son muy lentos para entrenar por el gran número de parámetros a ajustar y por la gran cantidad de imágenes de entrenamiento que se necesitan para lograr un correcto funcionamiento. Una de las primeras redes que podemos encontrar en la literatura fue utilizada para reconocer imágenes de dígitos manuscritos como los que se muestran en la figura 4.1. Para esto los autores entrenaron una red de 5 capas [LCDH$^+$90] a la que llamaron red de retro-propagación.

Pero relativamente hace pocos años con la aparición de grandes conjuntos de imágenes etiquetadas como ImageNet [DDS$^+$09] y la disponibilidad de Unidades de Procesamiento Gráfico

(GPU, por su denominación en inglés) muy potentes, estos tipos de modelos se convirtieron en el estándar de facto para aplicaciones de visión por computadora [SZ14]; aunque también se usan en otros ámbitos como procesamiento del lenguaje natural, identificación de drogas y otras aplicaciones de aprendizaje de máquina e inteligencia artificial [KSH12, LBH15].

La aplicación de estos modelos en problemas de AP aún es muy incipiente pero se pueden encontrar algunas en diversas áreas relacionadas con las plantas como clasificación [CAGL14], detección de enfermedades [SAA+16] o rasgos de estrés [SGSS16]. También se pueden encontrar ejemplos en la clasificación de uso de suelo, como en [KLSS17, JL16, ZMZ16, HKJ16]. En este capítulo se presenta una breve introducción a los modelos de CNN utilizados en esta tesis y también a la librería que se utiliza para el cómputo de los mismos.

Figura 4.1: Ejemplos de las imágenes de dígitos manuscritos usadas en una de las primeras redes profundas.

4.2. Fundamentos

Una CNN es un tipo especial de red neuronal profunda compuesta por el apilado de diferentes capas. Las primeras capas de estas redes aplican filtros de convolución con parámetros ajustables a través del entrenamiento, conocidas como capas de convolución y de ahí el nombre de este tipo de redes. En la figura 4.2 se muestra un filtro o núcleo de convolución de tamaño 3×3 aplicado sobre una entrada de tamaño 8×8. Entre estas capas además hay otras capas que realizan rectificaciones lineales y agrupamiento (*pooling*) de datos para ir reduciendo la dimensionalidad de la entrada y la complejidad de la red en general. Las primeras capas convolucionales son las responsables del reconocimiento de los detalles de bajo nivel de las imágenes como esquinas o bordes y las últimas son las responsables de detalles de alto nivel como partes de objetos o características del fondo [ZF14].

Luego de estas capas de convolución siguen otras capas, como las que normalmente se usan en redes neuronales, conocidas como capas totalmente conectadas. Estas últimas están compuestas por neuronas que están conectadas con todas las neuronas de las capas anteriores y se encargan de detectar y reconocer los patrones más abstractos y generar el resultado final.

Una red neuronal, compuesta por múltiples capas, puede ser vista como el cómputo de una composición de funciones en donde cada función representa una capa. La entrada, para el caso

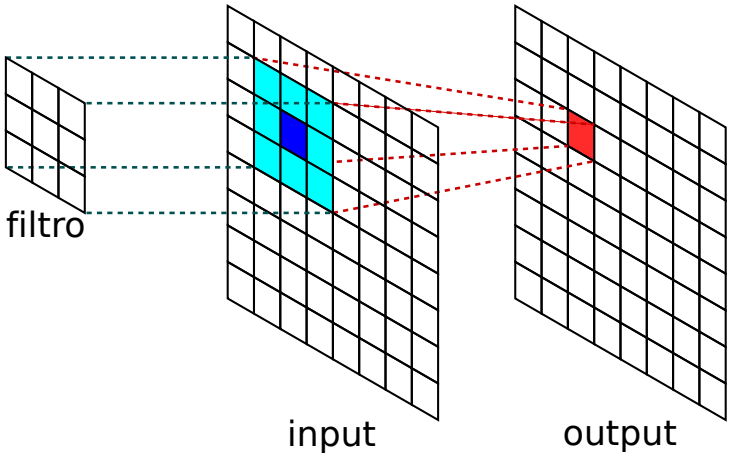


Figura 4.2: Ejemplo de un núcleo de convolución.

de clasificación de imágenes, es una imagen y la salida es un puntaje (score) de clasificación para cada una de las clases con la cual fue entrenada la red. Este puntaje representa cuanto "más probable" es que la imagen de entrada pertenezca a una categoría u otra.

Sean $\mathbf{X}$ los valores de los pixeles de la imagen de entrada, f_l la función que computa la capa l y $\mathbf{w}_l$ el vector de parámetros de dicha función, la salida de la CNN se obtiene como:

$$\mathbf{y} = f(\mathbf{X}) \stackrel{\text{def}}{=} f_L\left(\cdots f_2\left(f_1\left(\mathbf{X}; \mathbf{w}_1\right); \mathbf{w}_2\right); \mathbf{w}_L\right). \tag{4.1}$$

El vector de salida $\mathbf{y}$, es un vector que pertenece a $\mathbb{R}^c$ en donde c es el número de clases y cada componente del mismo representa cuanto más verosímil es que la entrada corresponda a la clase representada por esa componente. Luego utilizando la función argmax$(\mathbf{y})$ se computa cual es la clase que más se ajusta a la imagen de entrada.

Para ajustar los parámetros $\mathbf{w}$ de la red, $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_L)$, se debe minimizar una función de riesgo empírico utilizando un conjunto de entrenamiento. La función de riesgo a minimizar es:

$$L(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} l(z_i, f(\mathbf{X}, \mathbf{w})) \tag{4.2}$$

en donde $l(z, \hat{z})$ expresa la penalización de predecir la clase $\hat{z}$ en vez de z y N es el número de muestras. Para minimizar L se utiliza el algoritmo de descenso de gradiente:

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta_t \frac{df}{d\mathbf{w}}(\mathbf{w}^t) \tag{4.3}$$

Este algoritmo actualiza el peso actual $\mathbf{w}^t$ en la dirección de mayor descenso de la función de riesgo $L(\mathbf{w})$ computando su gradiente. En dicha ecuación η_t indica la tasa de aprendizaje.

4.3. Descripción de la red utilizada

Debido a la complejidad de estos tipos de redes, su diseño, prueba y puesta a punto requiere de grandes requerimientos computacionales y de tiempo. Como estas tareas no eran el objetivo de esta tesis se optó por utilizar una red ya diseñada y probada en diferentes experimentos y evaluar su comportamiento en problemas relacionados con la AP, tema poco explorado hasta el momento en la literatura.

Las áreas de investigación relacionadas con el DL y las CNN en particular están recibiendo una atención muy importante en los últimos años debido a los prometedores resultados que se obtienen en diferentes problemas de reconocimiento visual usando estos tipos de algoritmos. Una de las impulsoras de estos avances es ILSVRC (Competencia de Reconocimiento Visual de Gran Escala), en donde anualmente desde 2010 los investigadores evalúan y comparan algoritmos de detección de objetos y clasificación de imágenes en gran escala [RDS+15]. Entre las redes que mejores resultados obtuvieron en las diferentes ediciones de la competencia podemos nombrar a AlexNet [KSH12], VGG [SZ14], R-CNN [GDDM14], GoogLeNet [SLJ+], entre otras. Al comenzar los experimentos de esta tesis se decidió usar la CNN AlexNet porque era una de las pocas que estaba disponible en ese momento, además existían modelos ya entrenados en varias librerías (ver sección 4.6 para más detalles sobre la librería utilizada) y por último porque era la arquitectura de red usada para extraer descriptores como se propone en [RASC14]. Aunque durante el desarrollo de la tesis fueron apareciendo nuevas redes con arquitecturas diferentes (VGG, R-CNN, GoogLeNet, etc.) que iban superando a las anteriores, como no estuvo dentro de los objetivos investigar cual es la mejor arquitectura para este tipo de problemas, se decidió continuar usando AlexNet; pero queda abierto un análisis de las diferentes arquitecturas para ver cuales producen mejores resultados en problemas de AP que se caracterizan por grandes similitudes entre los conceptos a desambiguar, entre otras cosas.

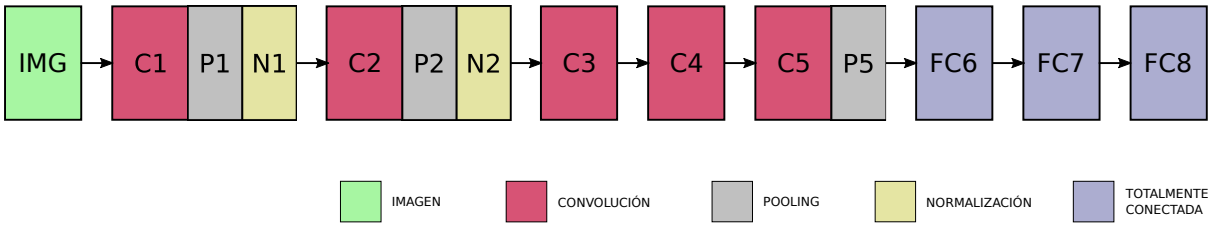


Figura 4.3: Diagrama simplificado de capas de la CNN utilizada en los experimentos.

En la figura 4.3 se ve un diagrama simplificado de la configuración de las distintas capas de la red que será usada en los experimentos del capítulo 5 y 6. Esta red, una de las CNN más utilizadas en la literatura, es conocida como AlexNet [KSH12] en honor a su creador y fue entrenada para clasificación en un subconjunto de 1 millón de imágenes de ImageNet las cuales pertenecen a 1000 diferentes clases de objetos.

La misma, compuesta por 650000 neuronas y con 60 millones de parámetros, está formada por 5 capas convolucionales, algunas de las cuales están seguidas por capas de agrupamiento y al final contiene 3 capas totalmente conectadas, la última de las cuales tiene 1000 salidas que se corresponden con las 1000 clases de entrenamiento. En la figura 4.4 se muestra la estructura completa de la red, con la definición de cada una de las capas. En dicha figura se puede apreciar que las capas de convolución están divididas en 2, las cuales permiten su entrenamiento en GPU diferentes. La entrada de la red es una imagen RGB de 224×244 pixeles y sobre ella se aplican los filtros de convolución de la capa C1, estos son 96 núcleos de tamaño $11 \times 11 \times 3$ con un paso de 4 pixeles. A la salida de esta, después de un paso de normalización y agrupamiento, se le aplica otra capa de convolución, C2, compuesta por 256 núcleos de tamaño $5 \times 5 \times 48$ cuya salida también es normalizada y agrupada. Las siguientes 3 capas de convolución, C3, C4 y C5 están conectadas entre sí sin ningún tipo de normalización ni agrupamiento. C3 está compuesta por 384 núcleos de tamaño $3 \times 3 \times 256$, C4 está compuesta por 384 núcleos de tamaño $3 \times 3 \times 192$ y la última capa convolucional, C5, está compuesta por 256 núcleos de tamaño $3 \times 3 \times 192$. La salida de C5, después de un paso de agrupamiento, está conectada a las capas totalmente conectadas.

Para el entrenamiento de esta red, los autores [KSH12] proponen la combinación de algunas técnicas ya existentes como el apagado de ciertas partes de la red para evitar el sobre-ajuste y la

propuesta de otras técnicas nuevas como la utilización de una función de regularización en las neuronas de la red conocida como función o unidad de rectificación lineal, la cual a diferencia de otras funciones permiten lograr un tiempo de entrenamiento 6 veces menor para un mismo error de entrenamiento. Además otra de las propuestas es dividir las primeras capas de la red en 2 partes como se muestra en la figura 4.4 y realizar el entrenamiento de cada una de estas partes en una GPU diferente permitiendo duplicar el tamaño del modelo gracias a la disponibilidad del doble de memoria.

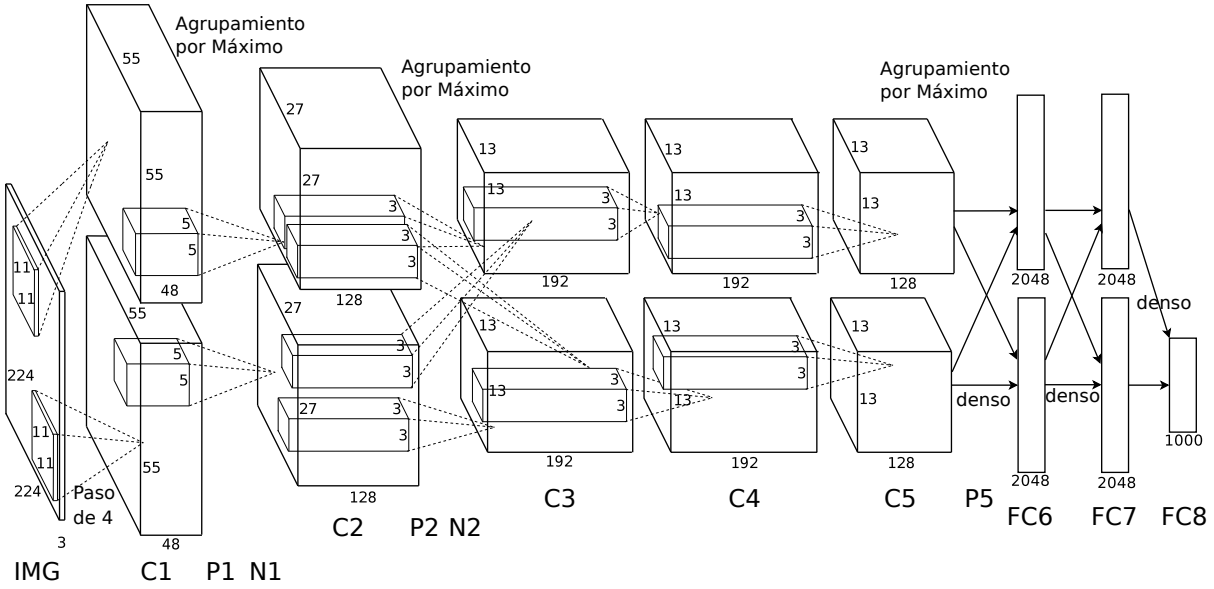


Figura 4.4: Estructura de la CNN utilizada, con la definición de cada una de las capas.

4.4. Ajuste fino de una CNN

Estos tipos de redes tienen alrededor de 100k parámetros, por lo tanto para su entrenamiento es necesario contar con un conjunto de entrenamiento como mínimo 10 veces mayor al número de parámetros. Como normalmente no poseemos esa cantidad de imágenes de entrenamiento sobre el problema a resolver y tampoco resulta factible construirlo, la solución normal es utilizar una red ya entrenada en otro conjunto y luego usando los datos específicos del problema, reentrenar solo algunas capas de la red, específicamente algunas de las capas totalmente conectadas. Este procedimiento se conoce como ajuste fino y se usará en el capítulo 6 para poder aplicar la red AlexNet a la clasificación de variedades de semillas de trigo. En la figura 4.5 se muestra un ejemplo de ajuste fino de una red. A la izquierda se muestran las últimas capas de la red AlexNet (figura 4.4) y a la derecha se muestra la misma red en donde se reemplazó la última capa para aplicarla a un nuevo problema. De esta manera, se utilizan la configuración y parámetros ya aprendidos de la primera red y se entrenan los parámetros de la nueva capa con las nuevas imágenes de entrenamiento. Esto permite entrenar una red muy grande usando pocas muestras de entrenamiento, con un bajo requerimiento de recursos computacionales pero con muy buena exactitud gracias a la reutilización de las capas ya entrenadas de la red.

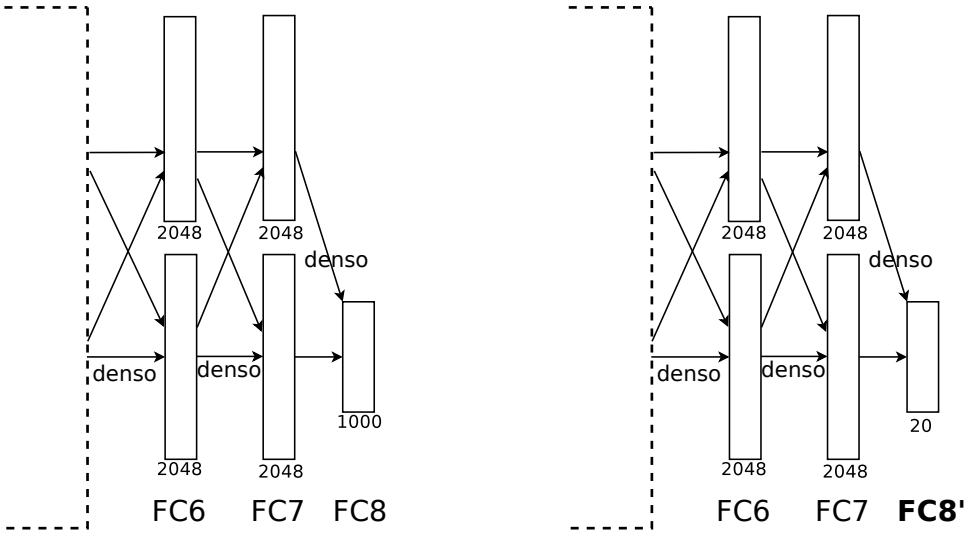


Figura 4.5: Ejemplo de la arquitectura de una red entrenada usando ajuste fino sobre otra ya existente. A la izquierda se muestra la red AlexNet y a la derecha una nueva red en donde se reemplaza la última capa de la red.

4.5. CNN como descriptores de características

Otra opción posible de uso de las CNN es para la extracción de descriptores en imágenes como se propone en [RASC14]. En dicho trabajo, los autores plantean usar una CNN ya entrenada para extraer descriptores visuales y luego aplicar estos descriptores a diferentes problemas de CV. Como ya se dijo, una CNN está compuesta por varias capas de las cuales las primeras son filtros de convolución más etapas de agrupamiento y normalización; estas primeras capas básicamente son filtros de convolución entrenados específicamente para reconocer las partes más importantes de las imágenes y descartar las no importantes, por lo tanto se puede decir que son "extractores de información". La red utilizada en ese trabajo es conocida como OverFeat [SEZ+14][1] y fue entrenada sobre ImageNet [DDS+09] para clasificar entre 1000 categorías de objetos. Los autores demuestran en [RASC14] que usando estos descriptores obtienen buenos resultados en problemas de clasificación de imágenes, reconocimiento de escenas, reconocimiento de grano fino, detección de atributos y recuperación de imágenes. Además estos experimentos fueron realizados sobre datasets con diferentes tipos de objetos como escenas interiores, escenas naturales, animales, objetos, aves y flores entre otros, mostrando la habilidad de generalización de estos descriptores.

Los descriptores calculados de esta manera, se pueden usar en forma similar a los descriptores comunes usados en la literatura como por ejemplo SIFT. La diferencia fundamental entre ambos tipos de descriptores es que la mayoría de los parámetros para el cómputo de descriptores del estilo SIFT son elegidos en forma manual y se mantienen fijos para todos los problemas considerados, en cambio los parámetros de los descriptores convolucionales no son fijados a mano, sino que son ajustados en grandes conjuntos de imágenes. Para la extracción de estos descriptores se puede utilizar la misma red que se muestra en la figura 4.3, pero tomando como salida alguna de las capas internas. Lo más común es tomar como descriptor la salida de la segunda capa totalmente conectada, FC7 en este caso. Estos descriptores serán referenciados en lo siguiente como CNNd. Como punto abierto de investigación, resta hacer un análisis ex-

[1] La arquitectura de la red OverFeat [SEZ+14] es muy similar a la red AlexNet [KSH12] y a los efectos prácticos podemos decir que son iguales [SEZ+14].

perimental del comportamiento de descriptores obtenidos con direfentes redes convolucionales ya entrenadas y utilizando la salida de diferentes capas como descriptores.

4.5.1. Clasificación usando descriptores CNNd

En esta tesis, estos descriptores son utilizados de dos formas, la primera es integrándolos al esquema de codificación eFV presentado en el capítulo anterior (ver figura 3.3) y la segunda es, como se propone en [RASC14], computando un descriptor CNNd global para la imagen y luego clasificarlo con un SVM lineal como se muestra en la figura 4.6.

Figura 4.6: Esquema de clasificación de CNNd usando SVM como se propone en [RASC14].

4.6. Librerías

Debibo a la complejidad computacional de los algoritmos involucrados y a las optimizaciones necesarias para obtener tiempos de cómputos razonables, se decidió utilizar una librería para realizar todos los experimentos relacionados con CNN. Los requerimientos fundamentales fueron:

- que permita ser usada con modelos ya entrenados debido a que este entrenamiento es muy costoso,
- que permita su uso para extraer descriptores de características como se menciona en la sección anterior,
- que permita realizar ajuste fino sobre una red ya entrenada,
- que tenga una interfaz en lenguaje Python para poder integrarla fácilmente a la librería **vrl** presentada en el capítulo anterior
- y por último que sea de código abierto para no tener que recaer en la compra de licencias como puede ser el caso de MATLAB.

Actualmente en la Wikipedia se listan alrededor de 40 librerías para resolver problemas de DL [2] lo que hace difícil hacer un análisis de todas ellas para elegir cual puede ser la que más se ajuste a las necesidades. Al momento de realizar los experimentos, aunque existían varias librerías maduras como Torch [CKF11] o Theano [BBB+10], la primera desarrollada en conjunto entre diversos centros de investigación y en la actualidad por varias empresas, entre ellas Facebook, y la segunda gestada en la Universidad de Montreal; las últimas innovaciones en el área DL se estaban publicando en conjunto con librerías específicamente desarrolladas para tal fin, como son el caso de OverFeat [3] [SEZ+14] o Caffe [JSD+14], entre otras.

Entre las librerías nombradas, con respecto a los requerimientos, aunque OverFeat es la librería propuesta para ser usada como extractor de descriptores en el trabajo [RASC14] esta es

[2] `https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software/Resources`

[3] OverFeat además de ser el nombre de una CNN, es una librería que permite realizar reconocimiento, localizaciń y detección usando dicha red.

Tabla 4.1: Comparación de las características de las distintas librerías analizadas.

Característica	Torch	Theano	OverFeat	Caffe
Extracción de descriptores	✓	✓	✓	✓
Ajuste fino de una red	✓	✓	✗	✓
Interfaz Python	✗	✓	✓	✓
Código abierto	✓	✓	✓	✓
Primera versión	2002	2009	2013	2014
Activa al 2018	✓	✗	✗	✓

la única que no permite ser usada para realizar el ajuste fino sobre una red ya entrenada por lo tanto se prefirió usar alguna de las otras. Siguiendo con el análisis, todas son de código abierto pero Torch es la única que en ese momento no tenía una interfaz para el lenguaje Python, aunque desde fines del 2016 existe PyTorch que permite utilizarla con código en Python aunque aún está en una versión muy temprana [4]. Fuera de los requerimientos, todas soportan paralelismo a nivel de CPU a través de OpenMP [DM98] y cómputo en tarjetas gráficas con CUDA [SK10]. En la tabla 4.1 se muestra un resumen de las características analizadas de las diferentes librerías.

Para esta tesis se decidió utilizar Caffe [JSD+14] porque cumplía con las necesidades planteadas y además era en ese momento una librería relativamente nueva, pensada desde un principio exclusivamente para aplicaciones de DL, a diferencia de otras más genéricas que están pensadas para ML o cómputo numérico como son Theano o Torch respectivamente, con lo cual sus algoritomos de bajo nivel están pensados específicamente para dichas tareas. Esta librería permite el entrenamiento, evaluación, ajuste fino y extracción de descriptores, con documentación y ejemplos para cada una de estas tareas. Como última ventaja, cuenta con una interfaz en Python que permite un prototipado rápido y la posibilidad de su inclusión en el código Python ya desarrollado en la librería **vrl** de una forma elegante y consistente.

Aunque en la actualidad, si habría que hacer nuevos experimentos, debido al gran avance en esta área del conocimiento se debería realizar un nuevo análisis de las distintas librerías existentes, porque en los últimos dos años han surgido muchas librerías y otras han quedado abandonadas. Una de las primeras que no tuvo más actualizaciones fue OverFeat. También los desarrolladores de Theano anunciaron hace poco que la versión 1.0 de fines del 2017 iba a ser la última versión [5] debido a que existían nuevas librerías desarrolladas por grandes empresas y centros de investigación y que por lo tanto seguir manteniendo está librería no era la mejor manera de contribuir al avance científico y la innovación. Con respecto a los nuevos desarrollos, ya nombramos a PyTorch que es una interfaz en Python para la librería Torch, pero aún se encuentra en una versión "beta". Otra de las librerías que evolucionó fue Caffe, debido a que Facebook contrató al desarrollador principal de esta para crear una nueva versión liberada en el año 2017 conocida como Caffe2 [6]. También la empresa Google liberó su motor de DL conocido como TensorFlow [7]. Por último, otra librería que apareció recientemente es MXNet [8] desarrollada por la fundación Apache y varias empresas más, entre ellas Intel.

[4] `http://pytorch.org/`
[5] https://github.com/Theano/Theano#mila-will-stop-developing-theano
[6] https://caffe2.ai/
[7] https://www.tensorflow.org/
[8] https://mxnet.apache.org/

Capítulo 5

Clasificación de imágenes de plantas

Índice

5.1. Resumen

En esta capítulo se presenta un análisis experimental del uso de los modelos presentados anteriormente, para resolver el problema de la identificación de plantas usando imágenes RGB. La tarea consiste en la identificación de la especie a la que pertenece una planta dada una imagen de alguna parte de la misma. Este problema presenta un desafío muy importante debido a la cantidad de especies y muchas veces a las similitudes entre distintas especies. En la sección experimental se compara el uso de diferentes descriptores con la codificación eFV y se evalúa la exactitud sobre conjuntos de datos públicos. Además los resultados obtenidos son comparados con métodos de estado del arte presentados en la literatura demostrando que la codificación eFV tiene un buen comportamiento para resolver este problema abriendo muchas perspectivas para aplicaciones reales.

5.2. Introducción

En los últimos años, ha habido un creciente interés en el problema de clasificación de especies de plantas usando información visual [CPB15, KT15, GJB+14, SBG14]. Algunas razones de esto son el gran número de especies en peligro de extinción y las altas tasas de deforestación debidas al corrimiento de la frontera agropecuaria y a un pobre planeamiento urbano. Las plantas tienen un rol crucial para la vida en la tierra y su descuido puede causar problemas irreversibles a la sociedad, como el calentamiento global, la pérdida de la biodiversidad y el daño ambiental [CPB15, WSM+14].

El problema presenta un desafío muy importante, debido a que es casi imposible de resolver para personas en general y muy difícil para personas entrenadas como granjeros, trabajadores

forestales e incluso botánicos [GBJ+13]. Las razones de esto son muchas, entre las cuales se puede nombrar el gran número de especies, numeradas en aproximadamente 200000, la gran variabilidad intraclase y la alta similitud entre especies diferentes [SBG14]. En la figura 5.1 se muestran ejemplos de hojas de distintas especies de plantas similares a las que se utilizarán en los experimentos.

Figura 5.1: Ejemplos de imágenes de hojas de distintas variedades de especies.

En este capítulo se aborda el problema usando el esquema eFV presentado anteriormente (capítulo 3). Como ya se mencionó, esta codificación es una generalización de la representación conocida como FV (capítulo 2), la cual permite la codificación de descriptores locales en un amplio dominio de entrada como vectores reales, enteros o binarios y matrices SPD.

El contenido de este capítulo está basado en los trabajos [RSP15b, RSP15a] presentados en el Simposio Argentino de Inteligencia Artificial (ASAI 2015) y en el Congreso Iberoamericano de Reconocimiento de Patrones (CIARP 2015).

5.3. Trabajos relacionados

Una considerable cantidad de literatura relacionada con el tema ha sido publicada, dentro de la que se incluyen algoritmos de preprocesamiento, extracción de características y clasificación específicos para este problema. Los diferentes enfoques pueden ser divididos en dos grandes grupos dependiendo del tipo de descriptores que utilizan, los que usan descriptores globales y los basados en descriptores locales.

Dentro de los primeros, los autores de [YAT12, YAY13] proponen el uso de descriptores globales de forma y textura obtenidos después de un paso de segmentación para la clasificación de imágenes de hojas. En [KT15] se propone un sistema que usa descriptores geométricos, matriz de distancia multiescala, momentos invariantes y un novedoso conjunto de descriptores globales. El cómputo de estos descriptores está basado en el contorno de la hoja, por lo tanto es necesario un paso de preprocesamiento para una correcta extracción del contorno y de acuerdo con los autores, este paso de extracción falla para algunas especies con hojas muy angostas como por ejemplo pinos. Un algoritmo semi-automático que devuelve las clases más probables en orden descendiente de confidencia es propuesto en [SBG14]. Los descriptores usados son globales y para su cálculo el usuario tiene que marcar la base y el ápice de la hoja. En [CPB15] se propone un método de reconocimiento basado en descriptores globales de forma y textura, los cuales son sensitivos a la rotación, translación y cambios de escala, por lo tanto antes de su extracción es necesario aplicar un algoritmo de alineamiento.

Con respecto a los métodos basados en descriptores locales, en [HKCL14] se propone un sistema basado en la codificación rala de descriptores SIFT y un método similar usando una combinación de descriptores incluyendo SIFT se propone en [PHG12]. En [BMOL+13], los autores proponen el uso de diferentes descriptores locales (SURF, Fourier, Rotacionales Invariantes, LBP) codificados con FV para clasificar imágenes de hojas tomadas con un fondo natural.

En ese trabajo los descriptores son computados sobre puntos de interés de Harris y clasificados con un SVM en una configuración uno contra todos. Los autores de [Nak13] usan descriptores locales (4 versiones de SIFT y auto-similitud) aumentados con un método polinomial que tiene en cuenta descriptores vecinos, luego estos son codificados con FV. En [CAGL14] se combina la codificación de descriptores SIFT y momentos de colores usando FV con CNN, incluyendo un paso de preprocesamiento para obtener la caja limitante más representativa de la imagen.

5.4. Descripción del método

Como ya se dijo más arriba, se propone resolver el problema de la clasificación de especies de plantas usando la codificación eFV que se presentó en el capítulo 3. Para esto se plantea utilizar un sistema de cuatro etapas, de manera similar al que se usó en la sección de experimentos del capítulo 3, la primera es la extracción densa de descriptores visuales, la reducción de estos descriptores usando PCA (solo para algunos descriptores, ver tabla 5.5), la codificación de los descriptores usando eFV y finalmente la clasificación de los vectores usando SVM. En la figura 3.3 se muestra un diagrama en bloques del mismo y a continuación se explica la configuración particular de cada una de estas partes.

5.4.1. Descriptores

Los descriptores son extraídos en forma densa sobre una grilla regular con el mismo paso en ambas direcciones. Además, estos son calculados en la imagen original en cuatro escalas, con un factor de escala de $\frac{1}{\sqrt{2}}$.

Los descriptores seleccionados para este trabajo son SIFT, Transformación de Características Invariantes ante Escala Binarizadas (BinSIFT, por su denominación en inglés), BRIEF, LBP y DCOV. PCA solo fue usado para los descriptores SIFT y para BinSIFT antes de la binarización.

5.4.2. Codificación eFV

Para los experimentos de este capítulo se utiliza la codificación eFV presentada anteriormente en la sección 3.6. En la tabla 5.1 muestra la distribución de la familia exponencial que se usa en la codificación para cada tipo de descriptor.

Tabla 5.1: Descriptores y su correspondiente distribución usada para la codificación.

Descriptor	Dominio de Entrada	Distribución
SIFT	$\mathbb{R}$	Gaussiana
BRIEF, BinSIFT, LBP	$\{0, 1\}$	Bernoulli
DCOV	Matrices SPD	Wishart

5.4.3. Clasificador

Para clasificar los eFV se usa una SVM con núcleo lineal entrenado con SGD, debido a que esta es la elección normal para este tipo de codificaciones [SPMV13, SR15]. El uso de núcleos no lineales es problemático debido a la muy alta dimensionalidad de los vectores.

5.5. Experimentos

Para evaluar la codificación, se realizan experimentos sobre diferentes conjuntos de datos públicos comúnmente usados para esta tarea y se comparan estos resultados con diferentes algoritmos del estado del arte.

5.5.1. Conjuntos de datos

Los cuatro conjuntos utilizados para los experimentos, están divididos por los creadores de los mismos en dos partes, una de entrenamiento y otra de evaluación. El primero de ellos es el presentado en [WBX+07], conocido como Flavia, el cual contiene 1907 imágenes de hojas de 32 clases de árboles, con un mínimo de 50 muestras por clase y un máximo de 72. El procedimiento normal de evaluación es dejar 10 muestras de cada clase para evaluación y el resto para entrenamiento, de esta manera se generan dos conjuntos, el de entrenamiento compuesto por 1587 (aproximadamente un 83 % de las muestras) muestras y el de evaluación con 320 muestras (aproximadamente un 17 % de las muestras). En la figura 5.2 se pueden ver ejemplos de las imágenes de este conjunto.

(a) Citrus reticulata (b) Prunus persica (c) Ligustrum lucidum

Figura 5.2: Ejemplos de imágenes de hojas del conjunto Flavia correspondientes a 3 especies comunes conocidas como mandarina, durazno y siempre verde respectivamente.

El segundo conjunto de datos usado es conocido como Foliage [KNSS11], el cual contiene 120 muestras para cada una de las 60 especies de árboles que contiene el conjunto de datos. El procedimiento recomendado de evaluación es tomar 100 muestras de cada clase para entrenamiento y 20 para prueba. En total el conjunto contiene 7200 muestras, la parte de entrenamiento contiene 6000 muestras (aproximadamente un 83 % de las muestras) y la de evaluación contiene 1200 muestras (17 % de las muestras).

Los otros dos conjuntos de datos, son los usados en la competencia de identificación de plantas organizada en ImageCLEF 2012 y 2013. Esta competencia, organizada en el marco de la conferencia CLEF, tiene como objetivo la clasificación de especies de plantas a través de imágenes de las mismas y una vez terminada pone a disposición del público en general los conjuntos de datos junto con sus anotaciones. Además, ambos conjuntos están divididos en 2 partes, entrenamiento y evaluación.

El primero de estos conjuntos, PlantCLEF2012 [GBJ+12], contiene 11572 imágenes de 126 especies de plantas divididas en tres tipos, “escaneadas”, “seudo-escaneadas” y “fotografías”. En las figuras 5.3, 5.4 y 5.5 se muestran ejemplos de hojas de las mismas especies pero con diferentes tipos de captura. Como se puede apreciar en las figuras, las hojas del tipo escaneadas contienen imágenes de hojas aplanadas capturadas con un fondo regular, las del tipo seudo-escaneadas son similares a las anteriores pero no se encuentran aplanadas de tal manera que pueden tener dobleces o sombras y las del tipo fotografías contiene fotos de hojas capturadas sobre un fondo natural, las cuales pueden tener distintos fondos, oclusiones u otros tipos de ruido. Estas características en las imágenes hacen que los algoritmos elegidos deban ser robustos ante

distintas variaciones para poder obtener resultados satisfactorios. El conjunto de entrenamiento contiene 4870 imágenes escaneadas, 1819 seudo-escaneadas y 1733 fotografías lo que hacen un total de 8422 imágenes (aproximadamente un 73 % del total). El conjunto de test contiene un total de 3150 imágenes (aproximadamente un 27 % del total), las cuales corresponden a 1760 imágenes escaneadas, 907 seudo-escaneadas y 483 fotografías. En la tabla 5.2 se muestra un resumen sobre la cantidad de imágenes que conforman este conjunto de datos.

Figura 5.3: Ejemplos de imágenes del tipo “escaneadas”.

Figura 5.4: Ejemplos de imágenes del tipo “seudo-escaneadas”.

Figura 5.5: Ejemplos de imágenes del tipo “fotografías”.

El segundo conjunto de la competencia, PlantCLEF2013 [GBJ+13], contiene 26077 imágenes de 250 especies de dos tipos, “hoja como fondo” y “fondo natural”. Las imágenes del tipo hoja como fondo contienen fotografías de la hoja de la planta, ubicada sobre una “hoja” de papel de color uniforme como fondo. Las del tipo con fondo natural contienen fotografías de diversas partes de plantas tomadas sobre un fondo natural, de manera similar al conjunto de datos anterior. Las partes que incluye este último tipo son cinco, imágenes de la “planta entera”, la “flor”, el “fruto”, la “hoja” y las “ramas”. En la figura 5.6 se muestran ejemplos de las imágenes para la especie *Kaki Persimmon*. El conjunto está dividido en dos partes, la de entrenamiento y la de evaluación. La de entrenamiento contiene 20985 (aproximadamente un 80 % del total) imágenes de las cuales 11204 corresponden al tipo hoja como fondo y 9781 corresponden a imágenes con el fondo natural; estás últimas a su vez se dividen en 1455 imágenes de la planta entera, 3522 de las flores, 1387 de los frutos, 2080 de las hojas y 1337 de las ramas. La de evaluación contiene 5092 (aproximadamente un 20 %) de las cuales 1250 son del tipo hoja como fondo y las restantes 3842 son del tipo fotografía; estás últimas están divididas en 694 imágenes de la planta entera, 1233 imágenes de las flores, 520 de los frutos, 790 de las hojas y 605 de las ramas. En las tablas 5.3 y 5.4 se muestra un resumen con la cantidad y porcentajes de imágenes de cada división y tipo para este conjunto de datos.

Tabla 5.2: Conjunto de datos PlantCLEF2012. Cantidad de imágenes de cada tipo para los conjuntos de entrenamiento y evaluación. Los porcentajes están redondeados a enteros.

Tipo	Entrenamiento	Evaluación	Total
Escaneadas	4870	1760	6630 (57 %)
Seudo-escaneadas	1819	907	2726 (24 %)
Fotografías	1733	483	2216 (19 %)
Total	8422 (73 %)	3150 (27 %)	11572 (100 %)

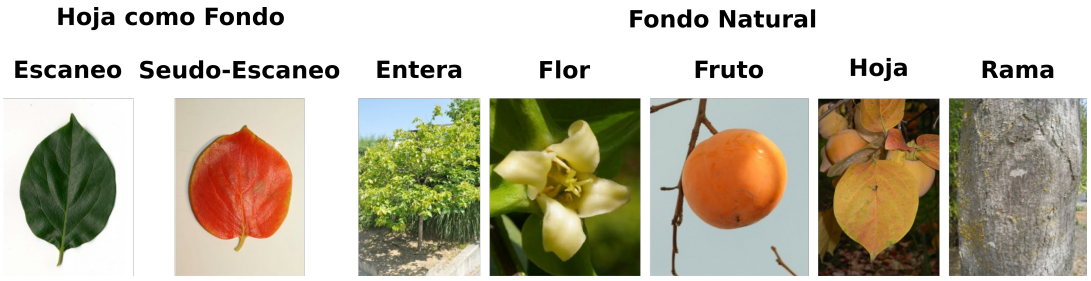


Figura 5.6: Ejemplos de los tipos de imágenes del conjunto PlantCLEF2013 para la especie *Kaki Persimmon*. Las primeras dos de la izquierda corresponden a las imágenes con una "hoja como fondo" y las últimas cinco corresponden a las que tienen un fondo natural.

5.5.2. Configuración experimental

Los descriptores locales se calculan sobre una grilla regular y en cuatro escalas con un factor de $\frac{1}{\sqrt{2}}$. En el caso de descriptores SIFT y BinSIFT, se redujo a 78 su dimensionalidad usando PCA. Sobre estos descriptores, se ajustó un modelo mezcla de la familia exponencial, el cual fue luego usado para calcular la codificación eFV, de acuerdo con la configuración mostrada en la tabla 5.1. En la tabla 5.5 se muestra un resumen de las diferentes configuraciones de eFV y su nombre corto para futura referencia. El cómputo de los eFV fue realizado con la librería **vrl** ya mencionada en el capítulo 3.

Además, se propone como una base de comparación el uso de descriptores computados desde CNN como se plantea en [RASC14] (ver capítulo 4 para mayor detalle). Como ya se dijo, estos descriptores fueron computados como la salida de la séptima capa (FC7) de la red neuronal presentada en [KSH12] conocida como AlexNet, cuya arquitectura se muestra en la figura 4.3. Luego para la clasificación de estos descriptores se usó un SVM. Este esquema de clasificación se puede apreciar en la figura 4.6. Esta línea de base de comparación será referenciada a continuación como CNNd+SVM [1]. Para mayor referencia ver el capítulo 4 en donde se realiza una pequeña introducción a las CNN, a la red utilizada en los experimentos de esta tesis y al uso de estas como descriptores de características.

[1]Hay que recordar que este método tiene una ventaja sobre el resto porque la red fue entrenada sobre un conjunto de $\approx 1M$ de imágenes dentro del cual se incluyen imágenes de plantas por lo tanto la comparación no es del todo justa.

Tabla 5.3: Conjunto de datos PlantCLEF2013. Cantidad de imágenes de cada tipo para los conjuntos de entrenamiento y evaluación. Los porcentajes están redondeados a enteros.

Tipo	Entrenamiento	Evaluación	Total
Hoja como fondo	11204	1250	12454 (48 %)
Fondo natural	9781	3842	13623 (52 %)
Total	20985 (80 %)	5092 (20 %)	26077(100 %)

Tabla 5.4: Subdivisión de las imágenes de tipo fondo natural para el conjunto de datos PlantCLEF2013. Cantidad de imágenes de cada tipo para los conjuntos de entrenamiento y evaluación.

Tipo	Entrenamiento	Evaluación	Total
Planta entera	1455	694	2149 (16 %)
Flor	3522	1233	4755 (35 %)
Fruto	1387	520	1907 (14 %)
Hoja	2080	790	2870 (21 %)
Rama	1337	605	1942 (14 %)
Total	9781 (72 %)	3842 (28 %)	13623 (100 %)

5.5.3. Ajuste de Parámetros

Para el cómputo de los eFV se debe ajustar el número de componentes K de la distribución mezcla (ecuación 3.6) y para la clasificación usando SVM se debe ajustar el valor del costo de clasificación C.

Con respecto al primero de ellos, aunque un mayor número de componentes produce un aumento en la exactitud, este viene acompañado de un mayor requerimiento de capacidades computacionales, principalmente memoria RAM y de un mayor tiempo de cómputo. Experimentalmente se lograron entrenar modelos de hasta $K = 1024$ restringidos a conjuntos pequeños de datos, pero se vió que el aumento en la exactitud era despreciable en relación al aumento en las capacidades computacionales requeridas, por lo tanto para estos experimentos se optó por $K = 256$, que permite tiempos de cómputo razonables, sin grandes requerimientos de memoria RAM y con una buena exactitud en los resultados.

Con respecto al ajuste del valor de costo del SVM, como los conjuntos de datos están formados por imágenes de contenido similar es probable que el valor óptimo no cambie mucho entre los diferentes conjuntos, además por la experiencia en otros conjuntos de datos se encontró que la exactitud no varía mucho para un rango determinado de este parámetro. Por estas razones y también para evitar realizar validación cruzada sobre conjuntos de datos tan grandes, se decidió ajustar este parámetro sobre el primer conjunto de datos presentado (Flavia) y utilizar el valor obtenido en el resto de los experimentos. El ajuste se realizó utilizando validación cruzada con 5 iteraciones sobre los datos de entrenamiento del conjunto de datos Flavia.

5.5.4. Resultados

En la tabla 5.6 se muestra la exactitud de las diferentes configuraciones del método propuesto sobre los conjuntos de datos Flavia y Foliage junto con resultados recientes disponibles en la literatura. La exactitud es computada como el porcentaje de muestras bien clasificadas del

Tabla 5.5: Configuraciones de eFV usados y abreviaciones.

Abreviación	Descriptor	PCA	Modelo mezcla
BRIEF-BMM-eFV	BRIEF	No	Bernoulli
SIFT-PCA-GMM-eFV	SIFT	Sí	Gaussiana
DCOV-WMM-eFV	Covarianza	No	Wishart
LBP-BMM-eFV	LBP	No	Bernoulli
BinSIFT-BMM-eFV	BinSIFT	Sí	Bernoulli

conjunto de evaluación sobre el total de muestras del mismo conjunto.

Tabla 5.6: Exactitud de las diferentes configuraciones de eFV y resultados en la literatura sobre los conjuntos de datos Flavia y Foliage.

Método	Exactitud Flavia	Exactitud Foliage
CNNd+SVM	99.06	**99.33**
SIFT-PCA-GMM-eFV	99.06	98.75
DCOV-WMM-eFV	**99.38**	98.25
LBP-BMM-eFV	95.62	93.25
BinSIFT-BMM-eFV	89.06	94.33
BRIEF-BMM-eFV	74.06	67.83
GLC [KT15]	93.00	-
SC [HKCL14]	95.47	-
CS [SBG14]	97.00	-
GLS [Kad14]	97.19	95.00
ICM [WSM$^+$14]	97.82	-

Como puede verse en la tabla 5.6, la mejor exactitud usando la codificación eFV es obtenida con descriptores SIFT y DCOV, y su exactitud en los conjuntos de datos Flavia y Foliage esta por encima de métodos recientes propuestos en la literatura. Además, en el conjunto de datos Foliage la línea de base tiene la mejor exactitud.

En las tablas 5.7 y 5.8 se comparan los resultados del algoritmo propuesto contra los mejores obtenidos en los desafiantes conjuntos de datos PlantCLEF2012 y PlantCLEF2013. El puntaje es un valor entre 0 y 1, siendo este último el mejor, y es computado usando los "scripts" provistos junto con los conjuntos de datos por los organizadores de la competencia. En negrita se resalta la mejor exactitud para cada tipo de imagen. En estos conjuntos de datos solo se muestra el puntaje para descriptores SIFT y DCOV.

En el conjunto de datos PlantCLEF2012 (tabla 5.7), la codificación de descriptores SIFT con eFV muestra la mejor puntuación para imágenes seudo-escaneadas, fotografías y en promedio; y el sistema de línea de base CNNd+SVM, muestra el mejor puntaje para las imágenes del tipo escaneadas.

Para el conjunto de datos PlantCLEF2013, el mejor puntaje para las imágenes de hoja como fondo es el obtenido por el método propuesto en [YAY13], pero este método falla para las imágenes con fondo natural como se puede ver en la tabla 5.8. La causa de este comportamiento es un paso de preprocesamiento que realiza una segmentación que es inaplicable para imágenes con fondo natural. Para este tipo de imágenes, uno de los mejores puntajes se obtiene con el método presentado en [Nak13] el cual está basado en un esquema complejo de fusión tardía de cuatro versiones de SIFT y descriptores de auto-similitud codificados con un embebido

Tabla 5.7: Resultados de clasificación sobre el conjunto de datos PlantCLEF2012 para los tres tipos de imágenes y en promedio.

Método	Escaneadas	Seudo-escaneadas	Fotografías	Promedio
CNNd+SVM	**0.65**	0.51	0.40	0.520
SIFT-PCA-GMM-eFV	0.62	**0.74**	**0.44**	**0.60**
DCOV-WMM-eFV	0.481	0.432	0.240	0.384
SABANCI OKAN [YAT12]	0.58	0.55	0.22	0.16
INRIA [BYM+12]	0.39	0.59	0.21	0.40
LSIS DYNI [PHG12]	0.41	0.42	0.32	0.42

polinomial de los descriptores previamente codificados con FV. Además, este último método usa información de meta-datos del conjunto de evaluación, en particular el tipo de imagen para el caso de las tomadas con fondo natural, en contraste con el propuesto que solo usa la información de la imagen. De nuevo, la línea de base CNNd+SVM, tiene el mejor puntaje para uno de los tipos de imágenes, lo cual nos habla del buen comportamiento de las redes convolucionales para ser usadas como descriptores de imágenes.

Tabla 5.8: Resultados de clasificación en el conjunto de datos PlantCLEF2013 para los dos tipos de imágenes.

Método	Hoja como fondo	Fondo natural
CNNd+SVM	0.557	**0.403**
SIFT-PCA-GMM-eFV	0.594	0.365
DCOV-WMM-eFV	0.363	0.181
SABANCI OKAN [YAY13]	**0.607**	0.181
NlabUTokio [Nak13]	0.502	0.393

5.6. Conclusiones y trabajo a futuro

En este capítulo se presentó una evaluación empírica detallada de diferentes configuraciones de eFV aplicada al problema de identificación de plantas. Los experimentos fueron realizados sobre diferentes conjuntos de datos públicos y se compararon los resultados de la codificación propuesta con los obtenidos con algoritmos de estado del arte, obteniendo para algunos casos resultados que son mejores que los presentados en la literatura. En la mayoría de los casos la mejor configuración es la codificación de descriptores SIFT con eFV, pero la línea de base usando CNNd y SVM también se comporta muy bien.

Las ventajas del método propuesto son que no se necesita un paso de preprocesamiento para la extracción del contorno debido a que está basado en descriptores locales, también permite el uso de diferentes descriptores en un esquema unificado, además no está basado en descriptores ajustados a mano o *ad-hoc* y es más simple que varios de los algoritmos existentes. Más allá de eso, a diferencia de otros métodos este puede ser aplicado sobre imágenes de hojas de plantas con un fondo simple o complejo como se demuestra en los experimentos.

En este capítulo se muestra que la codificación de descriptores SIFT con eFV es una buena opción para resolver el problema de identificación de plantas. También se muestra que los complejos descriptores CNNd funcionan bien y pueden ser una alternativa. Se puede concluir diciendo que el uso de nuevos modelos y técnicas actuales de CV basadas en ML producen

un salto en la exactitud en comparación con otros algoritmos basados en procesamiento de imágenes o soluciones heurísticas.

Como trabajo a futuro quedan abiertos varios experimentos que pueden mejorar la exactitud, dentro de los cuales destaco dos, el *primero* consiste en construir un esquema de clasificación usando la codificación de varios descriptores en conjunto con un esquema de fusión tardía de los resultados y el *segundo* consiste en codificar los descriptores obtenidos con las CNN usando eFV.

Capítulo 6

Clasificación de variedades de semillas de trigo

Índice

6.1. Resumen

En este capítulo se aborda el problema de la identificación de variedades de semillas de trigo. La identificación de semillas de cereales es una tarea realizada por personal calificado en diversas etapas de la producción agropecuaria, pero es una actividad lenta, tediosa y de baja repetibilidad. La disponibilidad de un método de clasificación automático de semillas acelera los procesos de evaluación y permite que sean realizados en diferentes etapas del proceso de producción de manera simple y con bajo costo. La solución propuesta, como se planteó en esta tesis, es el uso de técnicas actuales de clasificación de imágenes como son eFV y CNN. Con estas técnicas se logra una exactitud del 95 % en la clasificación de un conjunto de datos de semillas de 6 variedades de trigo recolectado para esta tarea. Dicho conjunto se encuentra disponible al público para futuras evaluaciones.

6.2. Introducción

El análisis de semillas es una actividad muy importante en el proceso de producción de granos. Esta actividad debe ser realizada en diferentes etapas del proceso, incluyendo la producción de semillas, la calificación del cereal para industrialización o comercialización y también durante las investigaciones científicas para mejoras de las especies [GNVC02]. También muchas veces por cuestiones legales, por ejemplo en la Argentina por reglamentación antes de la comercialización de distintos tipos de granos se debe realizar un análisis de una pequeña muestra del lote a comercializar [GVC05].

Esta tarea de identificación es realizada por personal capacitado usando inspección visual [LJS99], pero en la mayoría de los casos los métodos usados son lentos, tediosos [PKS+13], tienen baja reproducibilidad y agregan un grado de subjetividad difícil de cuantificar [GNVC02]. Esto se debe entre otras cosas a que el perito debe separar físicamente los granos e identificar el tipo de semilla [LJS99].

Esta actividad es clave para contribuir al agregado de valor al cultivo [GNVC02] y también porque el uso final de los granos depende del tipo y variedad de semilla específico [PP13]. En la actualidad el uso de semillas certificadas juega un rol muy importante en el incremento de la calidad y cantidad del cultivo [PPAFS12], por lo tanto antes de sembrar una variedad de semillas es muy importante confirmar la variedad a utilizar [PPAFS12]. El análisis también puede brindar conocimiento adicional sobre el proceso de producción, control de la calidad de las semillas y en la identificación de las impurezas [PKS+13].

Por lo dicho resulta de gran importancia técnica y económica la implementación de métodos automáticos basados en visión por computadora para una clasificación confiable y rápida de semillas [GNVC02]. Estos métodos también pueden ser explotados para detectar semillas infectadas con insectos o para detectar granos dañados [CM14].

En este capítulo se propone evaluar el comportamiento de las técnicas de visión por computadora ya presentadas en esta tesis, específicamente eFV y CNN, para resolver el problema de la identificación de la variedad de una semilla de trigo dada una imagen de la misma.

Estos algoritmos ya han sido usados, dando buenos resultados, en problemas de clasificación de imágenes en general como ImageNet [RDS+15] y en problemas de clasificación de subespecies, como por ejemplo aves [ZDGD14] o plantas (capítulo 5). En estos últimos tipos de problemas conocidos como de grano fino, al igual que el que se está tratando, las diferencias entre las clases son muy sutiles y escapan al ojo humano no entrenado. En la sección de resultados se demuestra que estos tipos de algoritmos se comportan muy bien en este problema en particular.

Las contribuciones de este capítulo son las siguientes: un sistema automático capaz de clasificar semillas de distintas variedades de una misma especie, en especial trigo, basado en técnicas actuales de clasificación de imágenes y un conjunto de imágenes de semillas de trigo con 6 variedades distintas. Esta es la primera vez que este tipo de técnicas son utilizadas para resolver este problema y también el conjunto recolectado es el primero en su tipo.

El contenido de este capítulo está basado en el trabajo [RGDPC16] el cual fue presentado en el Congreso de Agro-Informática (CAI 2016).

6.3. Trabajos relacionados

La propuesta es abordar el problema de la identificación de la variedad de una semilla de trigo dada una imagen de la misma, esta tarea es muy importante en diversas etapas del pro-

ceso de producción y obligatoria en otras. La evaluación es realizada por personal capacitado, pero es lenta, tediosa y de baja reproducibilidad, por lo tanto la posibilidad de automatizar este proceso puede hacer que sea más rápido y preciso, permitiendo una mejor determinación de la calidad de un lote y además que el mismo sea aplicado en varias etapas de la producción, lo cual le da un valor agregado al producto. También la identificación automática le permite a personas no entrenadas validar la procedencia de un lote de semillas en forma simple, por ejemplo con una aplicación corriendo sobre un teléfono móvil.

Existe bastante literatura sobre clasificación de semillas a través de imágenes, la cual se puede agrupar dependiendo del tipo de semillas, los descriptores que utilizan y el sensor con el cual se capturan las imágenes. En la mayoría de los trabajos, se utiliza un esquema de clasificación de imágenes general.

En algunos de ellos la clasificación se realiza entre semillas de distintas especies [MJ99, GVC05, PKS+13], entre semillas de una especie y contaminantes que pueden afectar la calidad de la misma [LJS99, RSJW15] y en otros la clasificación es entre variedades de la misma especie [WDL99, PPAFS12, PP13, CM14]. La clasificación de variedades de la misma especie es la que mayor dificultad tiene porque muestras de diferentes clases presentan una gran similitud, en cambio muestras de diferentes especies normalmente tienen una gran diferencia visual.

También se pueden encontrar diferentes sensores para adquirir las imágenes a clasificar, en la mayoría se usan cámaras RGB, pero en otros se utilizan cámaras NIR (Infrarrojo Cercano) [WDL99, RSJW15]. Con el uso de cámaras del tipo NIR se obtienen resultados levemente mejores, pero con la desventaja de que estas son menos comunes y más caras que las RGB.

Dependiendo del tipo de sensor, se pueden diferenciar a los trabajos según el tipo de descriptor que utilizan, para cámaras RGB los más usados son descriptores de color, morfológicos y de textura [LJS99, MJ99, GVC05, PPAFS12, PP13, PKS+13, CM14]. Para sensores NIR lo normal es utilizar descriptores obtenidos con técnicas espectrales [WDL99, RSJW15].

Una de las desventajas de las trabajos que se analizaron, a diferencia de la solución propuesta, es que en ninguno de ellos se pone a disposición para descarga el conjunto de datos con el cual se realizan los experimentos. La disponibilidad de estos datos permite que la comparación entre los distintos métodos sea más simple y rápida, contribuyendo al avance científico en el área.

6.4. Métodos

Para resolver el problema de identificación de la variedad de una semilla de trigo se plantea el uso de algoritmos actuales utilizados para clasificación de imágenes. Los algoritmos elegidos son eFV y CNN ya presentados anteriormente.

Como ya se dijo en el capítulo 3, un eFV es una representación global de una imagen que se obtiene agrupando características locales de una imagen (sección 6.4.1). Luego esta característica global, la cual es un vector, es usada como entrada a un clasificador lineal.

Las CNN, como se explicó en el capítulo 4, son un tipo de redes neuronales artificiales del tipo “feed-forward” en las cuales los patrones de conectividad entre las neuronas están inspirados en la organización de la corteza visual animal (sección 6.4.2).

6.4.1. Vectores de fisher de la familia exponencial

La representación usando eFV presentada en el capítulo 3 permite obtener un vector global que describe los descriptores locales de una imagen. La clasificación usando eFV contiene cuatro etapas, de manera similar a la usada en la sección de experimentos de los capítulos 3 y 5. La primera es la extracción de descriptores visuales de forma densa en la imagen, luego estos

descriptores son reducidos en dimensionalidad usando PCA [Bis06], estos nuevos descriptores son codificadas usando eFV y por último estos vectores son clasificados usando SVM lineales [FCH+08].

6.4.2. Redes neuronales convolucionales

Como ya se comentó en el capítulo 4 las CNN son variantes del perceptrón multicapa inspiradas biológicamente compuestas por el apilado de diferentes capas. Para un correcto funcionamiento se necesitan de muchas capas, por esa se habla de aprendizaje profundo, pero debido a esto, estos tipos de redes tienen alrededor de 100k parámetros, por lo tanto para su entrenamiento es necesario contar con un conjunto de datos de entrenamiento como mínimo 10 veces mayor al número de parámetros. Como normalmente no se cuenta con esa cantidad de imágenes de entrenamiento sobre el problema a resolver y tampoco resulta factible construirlo, la solución normal es utilizar una red ya entrenada en otro conjunto y luego usando los datos específico al problema, reentrenar solo algunas capas de la red, específicamente las capas totalmente conectadas. Esto se conoce como ajuste fino y se trató en la sección 4.4.

6.4.3. CNN como extractor de descriptores

Otra opción posible de uso de las CNN, como ya se expresó en el capítulo 4, es para la extracción de descriptores en imágenes [RASC14]. Para esto se utiliza la misma red que se muestra en la figuras 4.3 y 4.4, pero se toma como salida la segunda capa totalmente conectada, FC7 en este caso.

Estos descriptores son referenciados como CNNd y son usados de dos formas en este capítulo. La primera es para computar una base de clafisicación como se propone en [RASC14], como ya se hizo en el capítulo 5. Para esta línea de base se computa un descriptor CNNd sobre las imágenes y luego se entrena un SVM para clasificarlos (ver figura 4.6). La segunda forma de usar estos descriptores es codificándolos con el método de eFV que se explicó anteriormente. Estó se propuso como un trabajo a futuro, del capítulo 5, para la clasificación de plantas.

6.5. Experimentos

Para evaluar la factibilidad del esquema propuesto se realizaron experimentos con los diferentes algoritmos nombrados en la sección anterior sobre un conjunto de imágenes construido para esta tarea.

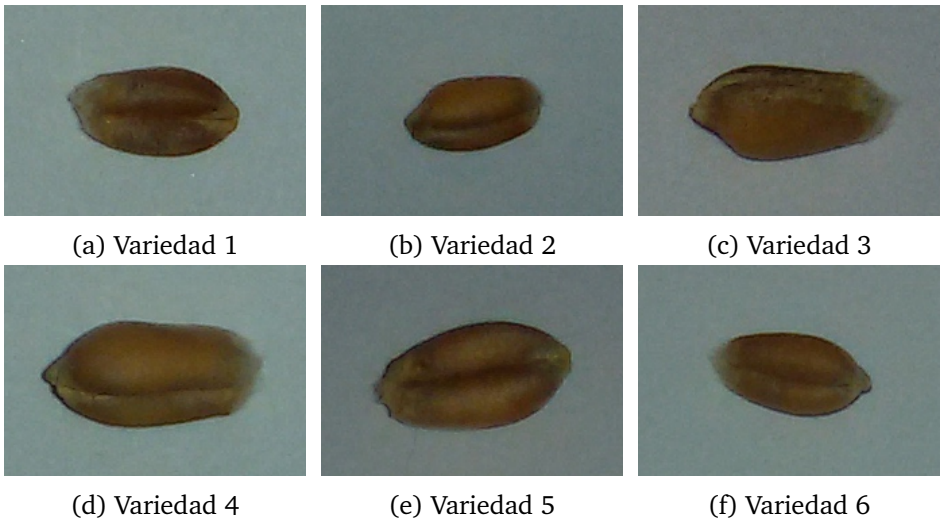
6.5.1. Conjunto de datos

Uno de los aportes de este trabajo es la construcción del conjunto de imágenes de semillas de trigo con su correspondiente etiquetado para su uso en la evaluación de algoritmos de clasificación. Dicho conjunto se encuentra disponible para descarga junto con una guía de usuario con el procedimiento recomendado para la evaluación y generación de los conjuntos de entrenamiento y evaluación [1]. Hasta lo que se conoce, este es el primero de este tipo disponible para descarga el cual permite realizar experimentos y comparar los resultados de los diferentes algoritmos en forma simple sin la necesidad de implementar los métodos a comparar. El mismo está compuesto por 315 muestras de 6 variedades diferentes de trigo las cuales se distinguen como variedad 1 a 6. Estas muestras (semillas) fueron clasificadas por profesionales en la materia. En la tabla 6.1 se muestra el número de muestras para cada variedad.

[1] http://ciii.frc.utn.edu.ar/JavierAndresRedolfi

Tabla 6.1: Cantidad de muestras por variedad.

Variedad	1	2	3	4	5	6	Total
# muestras	57	59	52	35	29	83	**315**



(a) Variedad 1 (b) Variedad 2 (c) Variedad 3

(d) Variedad 4 (e) Variedad 5 (f) Variedad 6

Figura 6.1: Muestra con las 6 variedades que componen el conjunto conjunto.

En la imagen 6.1 se muestran ejemplos de las 6 variedades que conforman el conjunto. Como se puede ver en la imagen las diferencias entre las variedades de semillas son casi imperceptibles para una persona no entrenada.

6.5.2. Configuración experimental

Como ya se marcó en la sección 6.4.3, la base de evaluación es la clasificación de descriptores CNNd con SVM, este método se designará como CNNd+SVM.

También se evalúa la clasificación usando eFV presentada en la sección 6.4.1 con diferentes descriptores. Los descriptores evaluados son SIFT [Low04], DCOV [TPM06] y CNNd (sección 6.4.3) los cuales serán designados SIFT+eFV, DCOV+eFV y CNNd+eFV respectivamente.

Por último como se plantea en la sección 6.4.2, se ajustan las últimas capas de la CNN AlexNet para este problema en particular; este método se indicará como CNN [2].

Para el entrenamiento de los diferentes algoritmos se dividió al conjunto en 2 partes, una de evaluación y otra de entrenamiento. Para ello, primero se enumeraron las imágenes de cada variedad en forma aleatoria y se seleccionaron las 20 primeras de cada una de las variedades para formar el conjunto de entrenamiento, las cuales hacen un total de 120 imágenes y las restantes 195 imágenes se usaron para evaluación.

6.5.3. Ajuste de Parámetros

Para los modelos basados en eFV igual que en el capítulo anterior se utilizaron mezclas con 256 componentes y el ajuste del parámetro C de los SVM utilizado para clasificar los eFV se realizó a través del algoritmo de validación cruzada con 5 iteraciones.

[2] Se vuelve a recalcar como en el capítulo anterior que este método tiene una ventaja sobre el resto porque la red fue entrenada sobre un conjunto de $\approx 1M$ de imágenes.

6.5.4. Código

Los cómputos relacionados con eFV se hicieron con la librería **vrl** [SR15] ya nombrada en el capítulo 3 y para los relacionados con CNN se utilizó la librería **Caffe** [JSD+14] presentada en el capítulo 4. Además para la facilidad de reproducción de los resultados, se ponen a disposición en la página web del autor los "scripts" necesarios para la generación de los mismos.

6.5.5. Resultados

Para comparar los diferentes métodos propuestos se evalúa la exactitud de cada uno de ellos en el conjunto de evaluación, el cual está compuesto por 195 muestras. La exactitud se computa como:

$$\text{Exactitud} = 100\,\frac{\text{Correctas}}{\text{Totales}} \tag{6.1}$$

en donde Correctas es el número de muestras de evaluación bien clasificadas y Totales es el número de muestras totales del conjunto de evaluación.

En la tabla 6.2 se pueden ver las comparaciones de la exactitud para cada método de clasificación propuesto y en la figura 6.2 su muestra un gráfico de la matriz de confusión normalizada para el método con el cual se obtuvo mayor exactitud (CNN).

Tabla 6.2: Resultados obtenidos con los diferentes métodos.

Método	**Exactitud (%)**
CNNd+SVM	83.59
DCOV+eFV	91.79
SIFT+eFV	81.54
CNNd+eFV	92.82
CNN	**95.42**

El método basado en CNN fue el que mejor se comportó para este problema superando en 8 puntos al mejor resultado publicado en la literatura [PP13], en el cual clasifican imágenes de 6 variedades de trigo, aunque este resultado no es directamente comparables debido a que se usa un conjunto de datos diferente y otro procedimiento de evaluación. Este resultado era esperado viendo el comportamiento de estas redes en otros problemas de visión por computadora [RASC14, Gir15] y clasificación de imágenes en particular [KSH12]. Aunque estas redes nunca habían sido utilizadas para este problema en particular.

Otra cosa interesante para hacer notar, es la mejora de 9 puntos en la exactitud que se obtiene codificando los descriptores CNNd con el esquema eFV o sin hacerlo, como se puede ver en la tabla 6.2 comparando la exactitud entre CNNd+eFV y CNNd+SVM, aunque esta mejora no es suficiente para superar a las CNN.

Con respecto a la exactitud de los diferentes métodos analizados, los que discriminan entre semillas de diferentes variedades o semillas de sus contaminantes, el cual es un problema más simple tienen una exactitud de entre 95 % a 100 %. En los trabajos en los cuales se discrimina entre variedades de la misma semilla [WDL99, PPAFS12, PP13, CM14], la exactitud es un poco menor y va entre 87 % a 98 %. Pero estos últimos trabajos, que obtienen una exactitud del 98 % no son directamente comparables con los que se obtienen en este capítulo porque en [WDL99] las imágenes son capturadas con una cámara NIR y en [CM14] no se clasifican imágenes de una

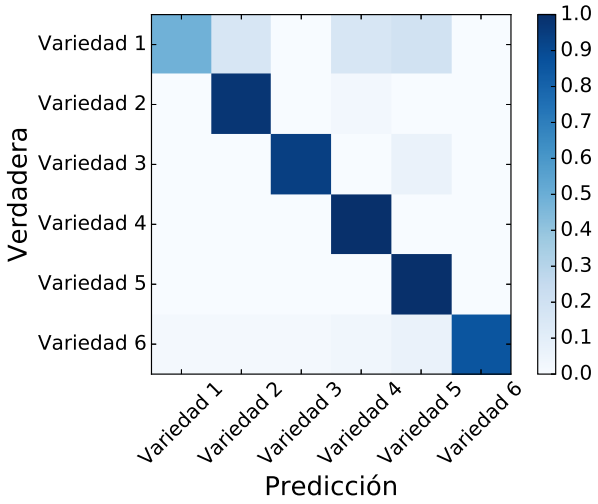


Figura 6.2: Matriz de confusión para la clasificación usando CNN.

sola semilla de trigo, sino que se clasifican imágenes con muchas semillas de trigo de la misma variedad.

6.6. Conclusiones y trabajo a futuro

En este capítulo se propuso el uso de las técnicas actuales de clasificación de imágenes y visión por computadora ya presentadas y aplicadas a lo largo de la tesis como son eFV y CNN para abordar el problema de identificación de variedades de semillas de trigo. Con el uso de CNN se logró una exactitud en la del 95 % lo cual demuestra la potencialidad de estos modelos. Además se contribuyó con un conjunto de imágenes de 6 variedades de trigo el cual se encuentra disponible para descarga.

Como trabajo a futuro se planea validar la capacidad del algoritmo para detectar defectos en las semillas como manchas, semillas partidas y también detectar partículas extrañas en las muestras, como podrían ser otras semillas, cáscaras, etc. Otra opción interesante es la detección de enfermedades y la estimación de la calidad de las muestras. Para evaluar estas nuevas propuestas es necesario aumentar el conjunto de imágenes con estos tipos de muestras. Por último, se está trabajando con biólogos de la universidad nacional de Córdoba en la identificación de dos clases de semillas de las sierras de Córdoba las cuales son muy parecidas visualmente, pero una de ellas es una especie nativa y la otra es una especie invasora o exótica que atenta contra la biodiversidad.

Capítulo 7

Clasificación de uso de suelo en imágenes PolSAR

Índice

7.1. Resumen

En este capítulo se estudia la aplicación de la codificación eFV presentada en el capítulo 3 al problema de clasificación a nivel de pixeles de imágenes PolSAR. Este es un problema desafiante ya que la información está codificada en forma de matrices de covarianza complejas, pero con el enfoque propuesto solo se considera la parte real de las covarianzas a nivel de pixeles, por lo tanto se sigue trabajando con matrices de covarianza pero reales. Aunque con esta aproximación se produce una pérdida de información los resultados obtenidos son alentadores. Primero se muestra que estas matrices preservan la propiedad de ser positivas definidas como su contraparte compleja y que se vuelven simétricas. Luego se aplica la codificación eFV derivada de mezclas de pdfs de Wishart. Resultados experimentales sobre dos conjuntos de datos desafiantes demuestran la efectividad del enfoque. Como trabajo a futuro se plantea la derivación de la codificación eFV con distribuciones complejas para poder aprovechar toda la información disponible, aunque esto está fuera de los objetivos de esta tesis.

7.2. Introducción

Un PolSAR es un dispositivo de sensado remoto capaz de proveer imágenes que son robustas a variaciones en la atmósfera y a las condiciones climáticas, sin importar la hora del día en las que son adquiridas. Esto las hace particularmente atractivas en aplicaciones tales como el monitoreo y el análisis automático de terrenos y cobertura del suelo [LP09, LGA+99, GYM14, WZT+16, JL16, ZMZ16, HKJ16]. Sin embargo, la mayor dificultad en el uso de datos PolSAR se origina en que la información (valores de los pixeles) está codificada como vectores o matrices complejas, dificultando la aplicación de técnicas estándar de la literatura de análisis estadístico y aprendizaje de máquina. Desde la perspectiva del modelado, tratar con este tipo de información de una manera fundamentada es un problema dificultoso que ha atraído una gran atención en el pasado.

Los datos PolSAR son generados transmitiendo pulsos electromagnéticos ortogonalmente polarizados hacia un objetivo y luego guardando el eco reflejado para cada canal en forma independiente. Las mediciones crudas son posteriormente procesadas para generar una imagen multicanal con valores complejos. Como consecuencia de la iluminación coherente, las imágenes son contaminadas con un tipo particular de ruido conocido como ruido moteado (speckle) [FNC11]. Para reducir el efecto de este ruido, los datos PolSAR son promediados en una pequeña vecindad, resultando en la llamada representación multivista de los datos PolSAR.

En este capítulo, se presenta una primera aproximación a la aplicación de modelos originados en la literatura de visión por computadora en conjunto con los modelos ya presentados en esta tesis al problema de clasificación de tipos de terreno, esto es la tarea de asignar etiquetas a pixeles basados en las propiedades de dispersión del objetivo medido por un sensor PolSAR. Concretamente, se propone un modelo que integra los formalismos de eFV presentados en el capítulo 3 con un modelo de energía basado en Potts [Pot52, BVZ01] que captura la dependencia espacial entre las variables. En el esquema de eFV, de manera similar a los experimentos presentados en capítulos anteriores, el contenido de la imagen (pixeles, regiones y/o toda la imagen) es caracterizado por el vector de gradiente normalizado derivado desde una distribución mezcla conveniente. En este caso, se considera la parte real de las covarianzas medidas por el sensor PolSAR y se deriva un eFV desde una mezcla de pdfs de Wishart reales demostrando primero que estas matrices son simétricas y que preservan la propiedad de definición positiva de su contraparte compleja. Luego se define un modelo de energía basado en Potts en donde los términos unitarios están computados como el negativo del producto interno entre los eFV por clase y los eFV a nivel de pixeles. La minimización de esta energía sobre el grafo de conexiones de 4 pixeles ubicados en la vecindad da la clasificación deseada.

Las principales contribuciones de este capítulo son un método novedoso basado en eFV para resolver el problema de clasificación de imágenes PolSAR (secciones 7.4.2 y 7.5). Hasta lo que se conoce, esta es la primera vez que la codificación eFV es aplicada al análisis de imágenes PolSAR. También se contribuyó un conjunto de anotaciones disponibles en forma pública y la definición de un procedimiento de entrenamiento y evaluación sobre dos conjunto de datos populares en la literatura (sección 7.6.1).

El contenido de este capítulo está basado en un trabajo el cual fue publicado en la revista Geoscience and Remote Sensing Letters [RSF17].

7.3. Trabajos relacionados

Uno de los primeros métodos propuestos para abordar este problema fue usar el Clasificador Complejo de Wishart (CWC, por su denominación en inglés) [LGA$^+$99]. El CWC está basado en ajustar una pdf compleja de Wishart, esto es una distribución sobre el conjunto de matrices hermitianas positiva definidas, por clase y clasificar los pixeles de acuerdo a una formulación de máximo a posteriori. Una de las mayores limitaciones del CWC es que al ajustar una sola pdf por clase hay una suposición subyacente de que los terrenos son homogéneos. Para tratar de superar esta limitación, en [GYM14] los autores usaron mezclas de distribuciones con un esquema de clasificación similar al anterior. Otros modelos de mezclas también han sido considerados en [DAE08, DE09] para el modelado y clasificación de datos PolSAR. En [DAE08], los autores llegaron a una generalización de la pdf de Wishart compleja analizando la distribución de la covarianza muestral para el modelo de mezclas de gaussianas escalado. En otro trabajo se propuso un modelo basado en la descomposición polar de la matriz de Mueller [WZT$^+$16] demostrando buenos resultados en la clasificación. Más recientemente, en [JL16] se propuso el uso de la red de apilado profundo de Wishart (W-DSN por sus siglas en inglés) la cual, como los modelos propuestos en [ZMZ16] y [HKJ16], tratan de aprovechar poderosas técnicas desarrolladas en la literatura de DL similares a las presentadas en el capítulo 4. Sin embargo, el entrenamiento de estos tipos de modelos requieren grandes cantidades de datos anotados los cuales son difíciles e imprácticos de obtener para este tipo de imágenes en particular.

7.4. Fundamentos

En esta sección se introducen los conceptos fundamentales de la generación de las imágenes PolSAR y de la representación usando eFV, ya descrita con profundidad en el capítulo 3. Para un tratamiento más profundo sobre imágenes PolSAR se recomienda al lector ver [LP09].

7.4.1. Datos PolSAR

Un SAR polarimétrico mide la señal de retrodispersión de un medio en cuatro diferentes combinaciones que resultan de transmitir y recibir la señal del radar con polarizaciones horizontales y verticales. La señal de dispersión es capturada por la siguiente matriz con elementos complejos:

$$\mathbf{S} = \begin{pmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{pmatrix}. \tag{7.1}$$

Para un medio recíproco, se cumple que $S_{HV} = S_{VH}$ y la información de dispersión puede ser alternativamente codificada como un vector:

$$\mathbf{h} = \begin{pmatrix} S_{HH} & \sqrt{2}S_{HV} & S_{VV} \end{pmatrix}^T \tag{7.2}$$

donde T denota transpuesta. La versión de múltiples vistas de este vector es obtenida promediando las mediciones individuales en una vecindad. Alternativamente se puede definir la covarianza compleja de múltiples vistas:

$$\mathbf{C} = [c_{ij}] = \frac{1}{m}\sum_{k=1}^{m} \mathbf{h}(k)\mathbf{h}(k)^*. \tag{7.3}$$

Aquí, $\mathbf{h}(k)$ es el vector de dispersión $\mathbf{h}$ en la posición k y el superíndice $*$ denota transpuesta conjugada. La matriz hermitiana $\mathbf{C}$ es Positiva Semi-Definida (PSD) y sigue una distribución de

Wishart con m grados de libertad [Goo63].

En este capítulo, en vez de trabajar con la matriz completa $\mathbf{C}$, solo se considera su parte real $\Re\{\mathbf{C}\} = [\Re\{c_{ij}\}]$. Para esto, primero se demuestra que $\Re\{\mathbf{C}\}$ es SPD y por lo tanto puede ser modelada por una distribución de Wishart real con n grados de libertad, con n un parámetro libre.

Se dice que una matriz $\mathbf{C} \in \mathbb{C}^{q\times q}$ es PSD si $\mathbf{z}^*\mathbf{C}\mathbf{z} \geq 0$, $\forall \mathbf{z} \in \mathbb{C}^q \setminus \{0\}$. Ya que $\mathbb{R}^q \subset \mathbb{C}^q$ se sigue que $\mathbf{x}^T\mathbf{C}\mathbf{x} \geq 0$, $\forall \mathbf{x} \in \mathbb{R}^q \setminus \{0\}$, es decir la matriz compleja $\mathbf{C}$ también es PSD en $\mathbb{R}^q$. Considerando el caso $m = 1$ en la ecuación (7.3) y denotando $\mathbf{h} = (h_1, \ldots, h_q)^T$. Ya que $\mathbf{C}$ es hermitiana, la forma bilinear de $\mathbf{x}^T\mathbf{C}\mathbf{x}$ sobre $\mathbb{R}^q$ puede ser escrita como:

$$\begin{aligned}\mathbf{x}^T\mathbf{C}\mathbf{x} &= \sum_i \mathbf{x}_i^2|h_i|^2 + \sum_i \sum_{j<i} \mathbf{x}_i\mathbf{x}_j\,(c_{ij} + \overline{c_{ij}}) \\ &= \sum_i \mathbf{x}_i^2\Re\{c_{ii}\} + 2\sum_i \sum_{j<i} \mathbf{x}_i\mathbf{x}_j\Re\{c_{ij}\} \\ &= \mathbf{x}^T\Re\{\mathbf{C}\}\mathbf{x} > 0.\end{aligned}$$

Sin embargo, notar que en general $\mathbf{x}^T\Re\{\mathbf{C}\}\mathbf{x} = \mathbf{x}^T\Re\{\mathbf{h}\mathbf{h}^*\}\mathbf{x} \neq \mathbf{x}^T\Re\{\mathbf{h}\}\Re\{\mathbf{h}\}^T\mathbf{x}$ y para el caso de múltiples vistas ($m > 1$ en la ecuación (7.3)) la matriz $\Re\{\mathbf{C}\}$ no puede ser considerada como antes, generada por una distribución de Wishart *con m grados de libertad*. En este caso, el número de grados de libertad es un parámetro que tiene que ser ajustado en forma empírica.

7.4.2. El principio de los eFV

Como ya se dijo en el capítulo 3 el eFV de $\mathbf{X}$ dado p_λ está definido como:

$$\mathcal{G}_\lambda(\mathbf{X}) \stackrel{\text{def}}{=} \frac{1}{N}\sum_{i=1}^{N} \mathbf{L}_\lambda \nabla_\lambda \log p_\lambda(\mathbf{x}_i). \tag{7.4}$$

La selección del modelo paramétrico p_λ depende de las particularidades del problema y en este caso se define p_λ como una mezcla de distribuciones sobre el conjunto de matrices de $q \times q$ simétricas positivas definidas, $\mathcal{S}(q)$, de la forma:

$$p_\lambda(\mathbf{x}) \stackrel{\text{def}}{=} \sum_{k=1}^{K} w_k p_k(\mathbf{x}), \tag{7.5}$$

con $\sum_{k=1}^{K} w_k = 1$, $w_k > 0$, $\forall k$ y p_k una pdf de Wishart con n grados de libertad parametrizada en la forma natural, esto es:

$$p_k(\mathbf{x}) = h(\mathbf{x})\exp(\langle \boldsymbol{\eta}_k, \mathbf{x}\rangle - \psi(\boldsymbol{\eta}_k)) \tag{7.6}$$

donde $h(x) = |\mathbf{x}|^{(n-q-1)/2}$, $\psi(\boldsymbol{\eta}_k) = \log\Gamma_q\left(\frac{n}{2}\right) - \frac{n}{2}\log|-\boldsymbol{\eta}_k|$, $\boldsymbol{\eta}_k \in \mathbb{R}^{q(q+1)/2}$ es el vector de parámetros naturales y $\Gamma_q(\cdot)$ es la función gamma multivariada.

Con la definición de arriba, el eFV de la muestra $\mathbf{X}$ dado por (7.4) es un vector de dimensionalidad $D = K\left(\frac{q(q+1)}{2} + 1\right)$. Como ya se dijo en el capítulo 3, se aplica la normalización de potencia y la normalización ℓ_2 al eFV ya que está demostrado que esto mejora el desempeño general en problemas de clasificación.

7.5. Clasificación basada en eFV

Para la clasificación se adopta un enfoque basado en un modelo de energía sobre un grafo G, en donde cada vértice o nodo representa un pixel de la imagen. Dicho modelo contiene dos términos, el primero que depende de los datos observados y el segundo que depende de la relación entre etiquetas vecinas, permitiendo darle más suavidad a la solución:

$$E(\mathrm{Y}) = E_{\text{datos}}(\mathrm{Y}) + E_{\text{suavizado}}(\mathrm{Y}) \tag{7.7}$$

donde Y es el etiquetado sobre el grafo $G = (\mathcal{V}, \mathcal{E})$ con vértices $i \in \mathcal{V}$ y aristas $(i, j) \in \mathcal{E}$. La energía que depende de los datos u observaciones analiza en forma individual a los pixeles, en cambio la otra tiene en cuenta la relación entre pixeles vecinos, penalizando a los pixeles adyacentes con diferentes etiquetas, logrando un suavizado en el resultado de clasificación.

Para esta función de suavizado, se elige un modelo simple de energía de Potts sobre el grafo de conexión de 4 pixeles vecinos, que corresponde a una vecindad horizontal y vertical de 3×3:

$$E(\mathrm{Y}) = \sum_{i \in |\mathcal{V}|} \phi_i(y_i) + \gamma \sum_{\{i,j\} \in \mathcal{E}} \mathbb{I}[i \neq j] \tag{7.8}$$

El primer término de la energía dependiente de los datos es el término $\phi_i(y_i)$, también conocido como potencial unario, que penaliza la incorrecta asignación de la etiqueta de clase y_i a la ubicación i. El segundo término, está compuesto por la función indicador $\mathbb{I}[z]$ que es igual a 1 si su predicado es verdadero y 0 en otro caso, y por la constante de penalización $\gamma \in \mathbb{R}$.

En la figura 7.1 se muestra la conexión entre un nodo central y sus vecinos, las aristas con líneas llenas indican la conexión del nodo con sus vecinos y la flecha indica el potencial unitario del nodo en cuestión. Como se puede observar cada nodo se conecta con cuatro de sus ocho vecinos.

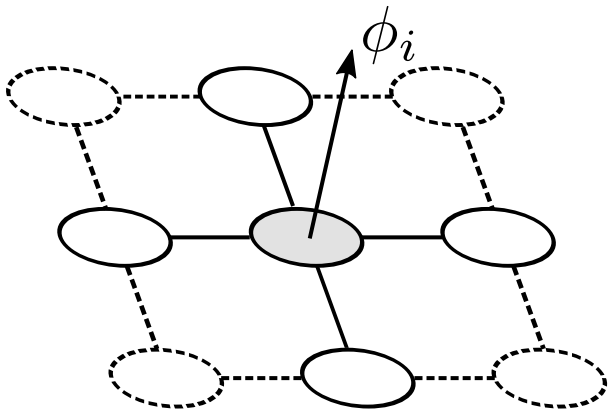


Figura 7.1: En la figura se muestra la conexión de un nodo del grafo con sus cuatro vecinos. Los nodos con líneas llenas son los que se conectan con el nodo central y las aristas en línea llena indican estas conexiones. La flecha indica el potencial unitario ϕ_i del nodo central.

Como el producto interno entre dos eFV es una medida de similitud entre ellos (distancia coseno), $\phi_i(y_i)$ está definido como el negativo del producto interno entre el eFV computado sobre una vecindad local a la ubicación i y un eFV por clase computado usando todas las muestras de entrenamiento para la clase $y_i \in \{1, \ldots, c\}$, esto es:

$$\phi_i(y) = -\left\langle \mathcal{G}_\lambda(\mathbf{X}_i), \mathcal{G}_\lambda^y \right\rangle \tag{7.9}$$

donde $\mathbf{X}_i$ es la muestra extraída de una vecindad de 3×3 vecinos centrados en la ubicación i y $\mathcal{G}_\lambda^y$ el eFV computado con las muestras de entrenamiento de la clase y.

El etiquetado que minimiza la energía $E(\mathrm{Y})$, $\hat{\mathrm{Y}} = \arg\min_{\mathrm{Y}} E(\mathrm{Y})$ da como resultado la clasificación deseada.

7.5.1. Minimización de la energía

Para minimizar la energía planteada en la ecuación 7.8 existen diversos algoritmos en la literatura, dentro de los que podemos destacar los basados en estimaciones de máximo a posteriori normalmente usados para minimización de energías de campos aleatorios de Markov [GG87]. Una de las desventajas de estos métodos es que las soluciones requieren de tiempos de cómputo exponenciales [BVZ01]. Debido a esto, para la minimización se recurrió a la solución aproximada presentada en [BVZ01] conocida como α-expansión o expansión-movimiento la cual está basada en gráficos de corte.

7.5.1.1. Algoritmo de expansión-movimiento

Este algoritmo es un método muy poderoso y ampliamente usado en la práctica. El algoritmo se basa en encontrar la expansión de alguna de las etiquetas que más hace disminuir la energía. Dicha minimización de la energía se resuelve usando gráficos de corte.

A continuación se muestra cuales son los pasos del algoritmo:

1. Iniciar con un etiquetado arbitrario.
2. Repetir para cada etiqueta:
   a) Encontrar el etiquetado de menor energía E con un movimiento de expansión usando gráficos de corte.
   b) Movernos a ese etiquetado si la energía es menor que la del etiquetado actual.
3. Si E no decrece en el ciclo, terminar el algoritmo, caso contrario volver a 2.

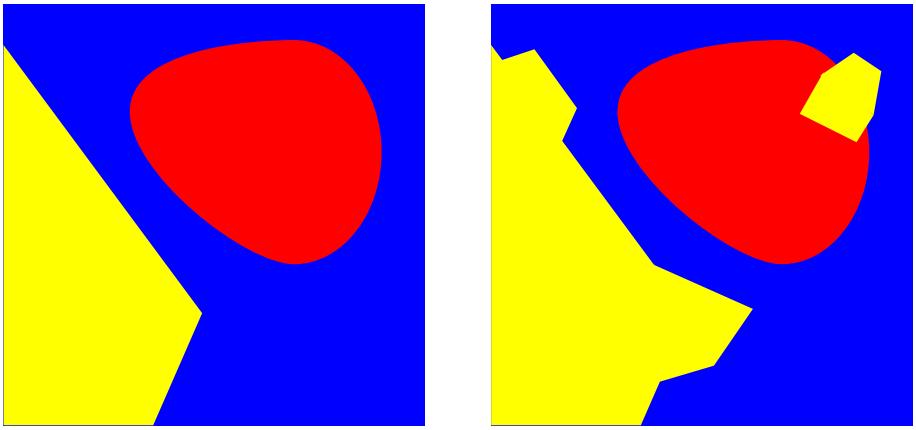
Las ventajas principales de este algoritmo es que tiene un tiempo de convergencia proporcional al número de nodos, a diferencia de otros algoritmos en las que suele ser exponencial. La desventaja fundamental es que este algoritmo no encuentra un mínimo global de la energía aunque sí encuentra un mínimo local y se demuestra que este mínimo local tiene una cota superior [BVZ01], que se muestra en la siguiente ecuación:

$$E(\hat{\mathrm{Y}}) \leq E(\mathrm{Y}^*) \leq 2\beta E(\hat{\mathrm{Y}}) \tag{7.10}$$

donde $\hat{\mathrm{Y}}$ es el etiquetado de menor energía global, Y^* es es mínimo local que encuentra el algoritmo y β es una constante que depende de la función de suavizado.

En la figura 7.2 se muestra un ejemplo del funcionamiento del algoritmo mencionado.

Obtención del etiquetado de menor energía. Para encontrar una expansión de alguna de las etiquetas (paso 2.a del algoritmo) se recurre al algoritmo de gráficos de corte. Para esto se transforma el problema en un grafo en donde el objetivo es encontrar el corte que divide al grafo en 2 partes con la mínima energía conocido como corte mínimo [BVZ01]. Se aclara que el gráfico de corte encuentra la división de mínima energía entre 2 etiquetas, por lo tanto para resolver un problema multi-etiqueta como es el caso, primero se selecciona una etiqueta y al resto de las etiquetas se las agrupa en una nueva etiqueta temporal. Esto se puede ver en la figura 7.3.



(a) Etiquetado en el paso i (b) Etiquetado en el paso $i + 1$

Figura 7.2: Ejemplo de un paso del algoritmo de expansión-movimiento. En este ejemplo tenemos 3 etiquetas; a la izquierda se muestra el etiquetado obtenido en el paso i y a la derecha se muestra el etiquetado para el paso $i + 1$ después de realizar un paso de expansión-movimiento de la etiqueta representada por el color amarillo.

(a) Etiquetado en el paso i (b) Etiquetado en el paso $i + 1$

Figura 7.3: Ejemplo de un paso del algoritmo de gráfico de corte. En este ejemplo las etiquetas azul y roja se convirtieron en una sola etiqueta; a la izquierda se muestra el etiquetado obtenido en el paso i y a la derecha se muestra el etiquetado para el paso $i+1$ después de realizar un paso de expansión-movimiento de la etiqueta representada por el color amarillo.

Después de esto, si la nueva energía obtenida para el paso $i+1$ es menor que la energía del paso i nos movemos a este nuevo etiquetado, como se indica en el paso 2.b, caso contrario terminamos el algoritmo.

7.6. Experimentos

Una de las mayores dificultades en la evaluación y comparación de los algoritmos de clasificación de imágenes PolSAR es la ausencia de un conjunto de prueba estándar y un procedimiento de evaluación como es común en otras áreas relacionadas con el análisis de imágenes. Por esto, primero se describen los datos y se propone un procedimiento de evaluación para los experimentos realizados. Luego, se evalúan diferentes aspectos del método planteado y se compara su desempeño contra el de otros métodos propuestos en la literatura.

Se hace notar que, en orden de facilitar la reproducibilidad, todos los datos y los "scripts" usados en los experimentos serán puestos a disposición en la web del proyecto [1].

7.6.1. Conjunto de datos

Para la evaluación, se considera un subconjunto de las imágenes que fueron puestas a disposición pública a través de PolSARpro [2] por la Agencia Espacial Europea más conocida como ESA. Este subconjunto consiste en dos imágenes polarimétricas completas en banda L adquiridas por el sensor AIRSAR de la NASA/JPL sobre el área de la bahía de San Francisco (SFB) en los Estados Unidos y sobre una región agrícola en la provincia de Flevoland en los Países Bajos. En las figuras 7.4 y 7.5 se muestra la representación en seudo-color de los datos polarimétricos (izquierda) y el etiquetado real (derecha) para las imágenes SFB y FL, respectivamente. Para SFB, se tomó de la imagen original de 900×1024 una parte de 500×500 pixeles ya que no existe el etiquetado verdadero para la imagen completa. Para la imagen de FL se consideró la imagen original, la cual tiene 750×1024. Las máscaras de segmentación para los dos conjuntos están basados en los trabajos [GYM14] y [AJE07], respectivamente.

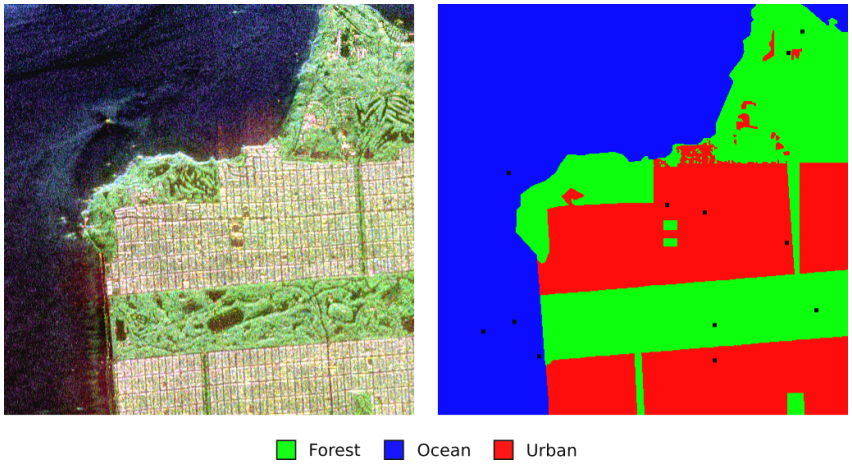


Figura 7.4: Imagen de la bahía de San Francisco (izquierda) y etiquetado con la verdad de suelo (derecha). Las muestras de entrenamiento están marcadas con cuadros negros (magnificar para apreciar mejor).

[1] http://ciii.frc.utn.edu.ar/JavierAndresRedolfi/sartb
[2] https://earth.esa.int/web/polsarpro

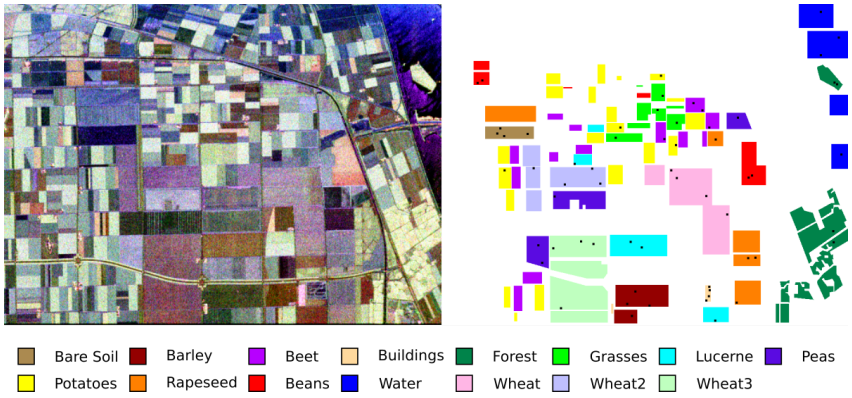


Figura 7.5: Imagen de Flevoland (izquierda) y etiquetado con la verdad de suelo (derecha). Las muestras de entrenamiento están marcadas con cuadros negros (magnificar para apreciar mejor).

Para la evaluación, se generaron 10 divisiones diferentes de entrenamiento/evaluación y se reporta la exactitud media (y desviación estándar) sobre las 10 corridas. Las divisiones son generadas siguiendo un procedimiento en el que se combinan dos estrategias comunes encontradas en la literatura, selección aleatoria [WZT+16, JL16, ZMZ16] y anotación manual de muestras de entrenamiento [ZWN+09, DAE08, ZSZM15]. El proceso es como sigue, para cada clase se seleccionan en forma aleatoria r ubicaciones de los pixeles etiquetados para esa clase y luego para cada una de las r ubicaciones se toma una pequeña ventana de un tamaño de $s \times s$ pixeles. Para forzar la variabilidad de los datos, se consideran solo ventanas que no se superpongan. Con este procedimiento, se termina con rs^2 muestras por clase que se usan para entrenamiento, mientras que el resto se usa para evaluación. En lo que sigue, se fija $r = 4$ y $s = 5$, lo que da un total de 100 muestras por clase. Además se ponen a disposición los "scripts" necesarios para generar las mismas divisiones que se usan para computar los resultados mostrados en este capítulo.

Con respecto al modo de selección de las muestras se hacen las siguientes observaciones. Primero, una muestra tomada de la forma descrita está compuesta por r subconjuntos de s^2 pixeles vecinos. De esta forma se mantienen algunas de las dependencias locales que existen entre pixeles en una ventana de tamaño $s \times s$. Estas dependencias son menos probables de reflejarse en muestras tomadas en forma aleatoria. Segundo, el proceso no está sesgado por causa de la subjetividad de la selección manual, favoreciendo la reproducibilidad y justicia al momento de comparar resultados.

La tabla 7.1 resume las principales características de los conjuntos de datos para esta particular elección de parámetros.

7.6.2. Detalles de implementación

Clasificar una imagen con este modelo involucra los siguientes pasos:

1. ajustar los parámetros de la distribución mezcla (7.5),
2. computar los eFV en cada ubicación de los pixeles y
3. resolver el problema de clasificación planteado en (7.8).

A continuación se dan detalles de los pasos mencionados. Primero, como las imágenes del conjunto de datos son el resultado de una sola pasada (vista simple), se deben convertir al formato de múltiples vistas, para esto se usa el software PolSARPro v4.2. Los parámetros de la

Tabla 7.1: Detalles de las imágenes PolSAR. Las dos últimas filas están computadas como promedio por clases.

	Bahía de San Francisco	Flevoland
Sensor	AIRSAR	AIRSAR
Banda	L	L
Filas	900	750
Columnas	1024	1024
Clases	3	13
Muestras	250000	176986
Muestras/Clase	83000	11799
Entrenamiento/Evaluación	0.0012	0.0085

distribución mezcla (7.5) son estimados con el algoritmo EM siguiendo un criterio de máxima verosimilitud desde el conjunto de muestras de entrenamiento (como máximo $1K$ muestras). En la práctica, solo se utiliza aquellas muestras cuyo determinante está dentro del percentil 95. Esto tiene el efecto de reducir el nivel de ruido durante la estimación sin tener que recurrir a filtros de suavizado o anti-moteado. Una vez que el modelo ha sido ajustado, los eFVs son computados a nivel de pixel como en [SR15]. Finalmente, el problema de inferencia asociado con la minimización de la ecuación (7.8) es resuelto con el algoritmo de grafos de corte usando la solución aproximada de [DOIB12], implementada en la librería GCO [3].

7.6.3. Selección de los parámetros n y K

En esta subsección se evalúa el impacto en el desempeño de los dos parámetros libres del modelo, estos son el número efectivo de grados de libertad n (ecuación (7.6)) y el número de componentes de la mezcla K (ecuación (7.5)). En la figura 7.6 se muestran los resultados obtenidos sobre el subconjunto SFB para diferentes valores de n y K. De la figura, se ve que K tiene la mayor influencia sobre la exactitud. El mejor desempeño es obtenido para $n \in \{5, 6\}$ y $K \in \{16\}$, aunque siendo muy estable dentro de esos rangos. En lo que sigue, se fija $n = 5$ y $K = 16$.

En las figuras 7.7 y 7.8 se muestran los resultados de la segmentación para las imágenes de SFB y FL respectivamente.

7.6.4. Comparación con otros métodos

Primero se compara el enfoque propuesto con dos métodos comunes en la literatura, estos son el clasificador complejo de Wishart (CWC) [LGA$^+$99] y un sistema similar al presentado en [GYM14] pero restringido a mezclas de pdfs de Wishart reales (RWM). En ambos casos, se ajusta un modelo para cada clase y se usa el negativo del logaritmo de la función de verosimilitud como potenciales unarios en la ecuación (7.8). La evaluación es llevada a cabo en los dos conjuntos de datos y los resultados se muestran en las figuras 7.9 y 7.10. De las figuras, se puede observar que el desempeño de RWM para todos los valores de K es similar al de CWC, con el modelo propuesto (eFV) superando a ambos algoritmos para $K > 2$. La gran variabilidad observada con CWC y RWM puede ser atribuida al bajo número de muestras disponibles para poder aprender

[3] `http://vision.csd.uwo.ca/code/`

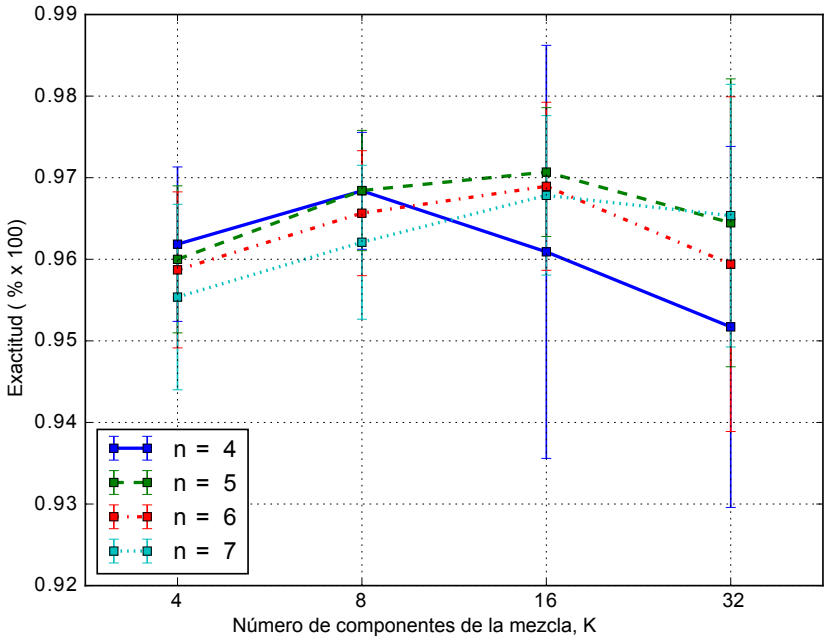


Figura 7.6: Exactitud media y desviación estándar medida en el subconjunto SFB para diferentes elecciones de n y K.

Figura 7.7: Resultado de la segmentación sobre SFB usando los parámetros seleccionados.

un modelo en forma apropiada. En el caso de eFV, la pdf subyacente no es específica para cada clase y es aprendida desde un conjunto más diverso y grande de muestras.

A continuación, se compara el modelo contra otros dos métodos recientemente propuestos en la literatura [4]. El primero es un método basado en la descomposición polar de la matriz de Mueller [WZT+16] mientras que el segundo corresponde a la red de apilado profundo de

[4]No se comparan los resultados contra los presentados en [GYM14] debido a que en dicho trabajo no se explica correctamente la implementación del método de clasificación ni el procedimientdo de evaluación.

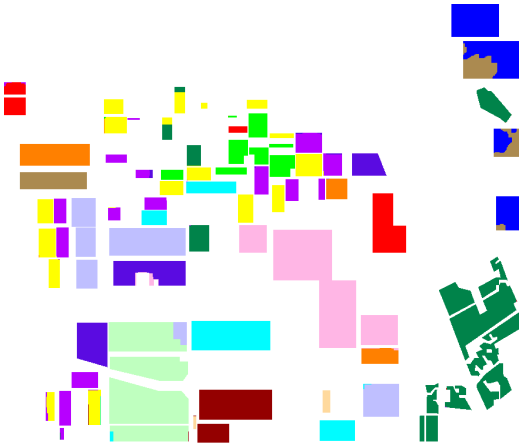


Figura 7.8: Resultado de la segmentación sobre FL usando los parámetros seleccionados.

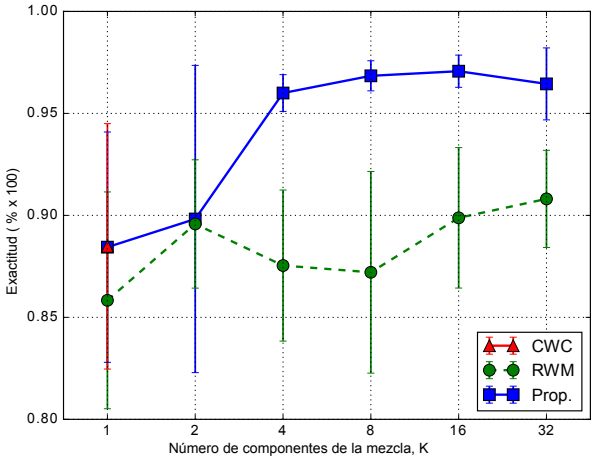


Figura 7.9: Exactitud sobre SFB para diferentes modelos y componentes de la mezcla, K. Para $K = 1$ la exactitud del método propuesto es similar a la obtenida con el CWC por esto no se aprecia correctamente en la figura.

Tabla 7.2: Comparación con el mejor de los métodos propuestos en [WZT$^+$16] sobre FL.

Método	BareSoil	Beet	Forest	Grasses	Lucerne	Peas
D-R-A$_\Delta$-Δ-m$_{00}$ [WZT$^+$16]	0.9878	0.9371	0.9496	0.8489	0.9231	0.9555
Propuesto ($K = 16, n = 6$)	1.0	0.9926	0.9770	0.9920	0.9261	0.9993
	Potatoes	Rapeseed	Beans	Water	Wheat	Promedio
	0.8896	0.9486	0.9653	0.9642	0.8864	0.9324
	0.9959	0.9983	0.9757	0.8472	0.9971	0.9728

Wishart (W-DSN) de [JL16]. La evaluación es realizada solamente sobre el subconjunto FL porque en ambos trabajos solo se reportan resultados sobre este. En el primer caso, siguiendo la configuración de [WZT$^+$16], se reportan resultados sobre 11 de las 15 clases y se entrenan los modelos usando 2000 muestras aleatorias por clase para una justa comparación. Los resultados

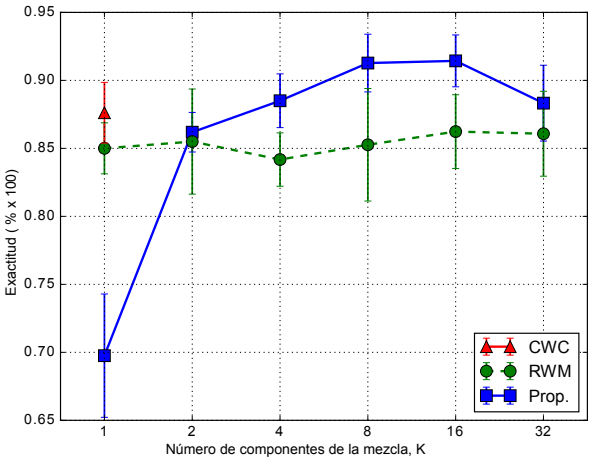


Figura 7.10: Exactitud sobre FL para diferentes modelos y componentes de la mezcla, K.

usando eFV junto con los reportados en [WZT+16] para el sistema D-R-A_{Δ}-Δ-m_{00} se muestran en la tabla 7.2. De esta, se puede ver que el modelo propuesto se comporta mejor en todas las clases excepto en la clase *Water* y que la exactitud media ($0{,}9728$) está 4 % por encima de la del otro método.

También se compara el modelo con el enfoque basado en DL de [JL16]. En este caso, el desempeño en clasificación se reporta como la exactitud media sobre las 15 clases, usando para entrenamiento el 5 % de las muestras etiquetadas para cada clase y el restante 95 % para evaluación. El sistema W-DSN logró una exactitud promedio (sobre las 15 clases) de $0{,}9268$ mientras que con el propuesto se logró una exactitud de $0{,}9688$, lo cual representa una ganancia de un 4 % en comparación con W-DSN.

7.7. Conclusiones y trabajo a futuro

En este capítulo se presentó un estudio preliminar sobre la aplicación de eFV al problema de clasificación de imágenes PolSAR obteniendo resultados alentadores sobre dos conjuntos de datos reales usados comúnmente en la literatura.

Como trabajo a futuro, una extensión natural de este modelo es el uso de mezclas de pdfs de Wishart complejas o modelos más complejos adaptados específicamente a datos PolSAR [FFC05]. Para la primera extensión hay que desarrollar una nueva formulación de eFV en donde se contemplen distribuciones de probabilidad del tipo complejo. Estas líneas de investigación serán perseguidas en trabajos futuros.

Capítulo 8

Evaluación final

Índice

La hipótesis fundamental de esta tesis es que la integración de modelos de clasificación y recuperación de imágenes recientemente propuestos en la literatura (específicamente FV y CNN) a problemas de AP otorgará un mayor nivel de robustez y escalabilidad a los sistemas actuales. También permitirá el abordaje de otros problemas de AP que hasta el momento no habían sido solucionados con este tipo de herramientas. Además se presume que el desarrollo de nuevos modelos en base a los existentes, en particular la extensión de FV, acentuará estos beneficios. Hasta donde se sabe, el uso de estos modelos ha sido poco explorado en problemas de AP o su aplicación es muy incipiente.

8.1. Conclusiones

En la actualidad existen dos tipos de modelos fundamentales para resolver problemas de visión por computadora, estos son los basados en BoVW y los basados en CNN. Dentro de los primeros el más usado es la codificación usando FV. En base a esta última, en esta tesis se desarrolló un modelo conocido como eFV el cual es una generalización del primero que permite el abordaje de una cantidad mayor de problemas con un sustento teórico firme. Una vez desarrollados estos modelos, los cuales fueron aplicados en problemas generales como clasificación de imágenes, se validaron para su aplicación en problemas de AP. Como primer problema

se planteó el de clasificación de imágenes de plantas, el cual hasta el momento solo había sido abordado con herramientas bastante alejadas de los esquemas actuales de clasificación. Siguiendo este camino se evaluó el problema de clasificación de variedades de semillas de una misma especie (trigo), problema más difícil aún que el anterior debido a la gran similitud visual entre los conceptos a desambiguar. Por último, para validar la versatilidad de aplicaciones del modelo desarrollado, se planteó la clasificación de uso de suelo en terrenos a través de imágenes PolSAR. Esta última aplicación es muy diferente a las demás debido a que las imágenes no son obtenidas con cámaras RGB, sino que son capturas con radares activos lo cual hace que la naturaleza de los datos sea muy distinta, además de ser capturados a una distancia muy grande del objetivo, lo que genera la presencia de grandes niveles de ruido.

En la sección experimental de esta tesis se pudo comprobar a través de experimentos muy variados que el uso de técnicas actuales y el desarrollo de nuevos modelos produce un aumento muy importante en la exactitud de los algoritmos evaluados. Además le otorga robustez contra problemas como el ruido como se pudo ver en los problemas de clasificación de plantas en donde aparecen imágenes con sombras, oclusiones, etc. o de imágenes PolSAR las cuales sufren del grave problema del ruido moteado. También estas técnicas demostraron un buen comportamiento en problemas de grano fino, en donde existen grandes similitudes en los procesos a evaluar como se pudo ver en el problema de clasificación de semillas. Por último se demostró que al extender los dominios de entrada de los algoritmos se pueden abordar un número mayor de problemas de manera fundamentada sin tener que recurrir a heurísticas.

8.2. Contribuciones

A modo de resumen, en esta sección se establecen las principales contribuciones realizadas en esta tesis.

Como punto de partida, en el capítulo 3 se presentó un formalismo para la codificación de imágenes que extiende FV a mezclas de pdfs no gaussianas. Este modelo provee una estructura unificada para la representación de imágenes usando características locales definidas sobre dominios de entrada generales. El modelo fue evaluado empíricamente sobre conjuntos de datos generales, para modelos basados en mezclas de pdfs gaussianas, Bernoulli, Wishart y Dirichlet. Los resultados muestran la gran flexibilidad y el poder de modelado del enfoque propuesto. [SR15]

Luego se presentó en el capítulo 5 una evaluación empírica detallada de diferentes configuraciones de eFV aplicada al problema de identificación de plantas. Los experimentos fueron realizados sobre diferentes conjuntos de datos públicos y se compararon los resultados con diferentes algoritmos de estado del arte obteniendo, para algunos casos, resultados que son mejores que los presentados en la literatura. En la mayoría de los casos la mejor configuración es la codificación de descriptores SIFT con eFV, pero la línea de base usando CNN y SVM también se comporta muy bien. [RSP15a]

En el capítulo 6 se propuso el uso de estas técnicas para abordar el problema de identificación de variedades de semillas de trigo, el cual hasta el momento solo había sido abordado con técnicas que se pueden considerar como “muy manuales”. Con el uso de CNN se logró una exactitud en la identificación del 95 % lo cual demuestra la potencialidad de estos modelos. Además se contribuyó con un conjunto de datos de 6 variedades de trigo el cual se encuentra disponible para descarga. Este último aporte también es muy importante porque permite que

otros autores tengan una base de imágenes etiquetadas para la comparación de los algoritmos evitando el proceso engorroso de la captura y clasificación de las mismas. [RGDPC16]

Por último en el capítulo 7, se propuso un esquema novedoso que integra los formalismos de eFV presentados en el capítulo 3 con un modelo de energía basado en Potts, para la clasificación de tipos de terreno usando imágenes PolSAR. Con este esquema se obtuvieron resultados alentadores sobre dos conjuntos de datos usados comúnmente en la literatura. Hasta lo que se conoce, esta es la primera vez que la codificación FV o su generalización eFV es aplicada al análisis de imágenes PolSAR. Además se construyó un conjunto de anotaciones disponibles en forma pública y la definición de un procedimiento de entrenamiento y evaluación sobre dos conjunto de datos populares en la literatura. De esta manera en un futuro será más fácil la evaluación y comparación de algoritmos en esta aplicación. [RSF17]

8.3. Perspectivas

Viendo la gran potencialidad de los eFV para su aplicación en problemas de AP, y en otros también, se pretende continuar esta tesis con la generalización de estos modelos a dominios de entrada aún más amplios. Una extensión natural es el uso de mezclas de pdfs complejas, la cual para el caso de Wishart tiene una aplicación directa en la clasificación de imágenes PolSAR.

Con respecto a la clasificación de plantas, en la actualidad hay un problema muy grave por las grandes cantidades de pesticidas utilizados en los procesos agrícolas tanto intensivos como extensivos. La optimización del esquema planteado en el capítulo 5 para su aplicación en condiciones de campo y en tiempo real, puede servir de puntapié para la mejora de los sistemas de aplicación selectiva de agroquímicos. También estos sistemas pueden ser utilizados para realizar mapas de población de plantas que permitan realizar un seguimiento del estado y evolución de la vegetación y del uso del suelo.

También con respecto a la clasificación de semillas, el sistema presentado puede ser mejorado para darle la capacidad de detectar defectos en las mismas como pueden ser manchas, semillas partidas y/o también detectar partículas extrañas en las muestras, como podrían ser otras semillas, cáscaras, etc. Otra opción interesante es dotarlo de la capacidad de detectar enfermedades y de estimar la calidad de las muestras. Para estas nuevas propuestas se planea aumentar el conjunto de datos con imágenes de semillas enfermas y también contaminadas con otros elementos. Además actualmente se está trabajando con biólogos de la universidad nacional de Córdoba en la aplicación de este esquema para la discriminación entre dos semillas de plantas que se pueden encontrar en las sierras de Córdoba, las cuales son muy similares visualmente pero con la salvedad de que una de ellas es de una especie nativa y la otra es de una especie implantada o invasora que atenta contra el bosque nativo.

En resumen, hay varios desafíos que quedan abiertos después del desarrollo de esta tesis. Tanto desde un punto de vista teórico, como es el desarrollo de nuevos modelos, pero también desde el punto de vista de las aplicaciones. Estas líneas de investigación serán perseguidas en trabajos futuros.

Apéndice A

Demostración de porque la FIM es semidefinida positiva

En este apéndice se demuestra que la FIM es una matriz semidefinida positiva. Sea I_λ la FIM definida como en la ecuación (2.2):

$$\mathbf{I}_\lambda = E_{\mathbf{x}\sim p_\lambda}\left[G_\lambda(\mathbf{X})G_\lambda(\mathbf{X})^T\right] \qquad .$$

Sea $\mathbf{A}$ una matriz cuadrada que pertenece a $\mathbb{R}^{d\times d}$ y $\mathbf{u}$ un vector no nulo que pertenece a $\mathbb{R}^d$, por definición se dice que una matriz $\mathbf{A}$ es semidefinida positiva cuando:

$$\mathbf{u}^T\mathbf{A}\mathbf{u} \geq 0 \qquad ,$$

reemplazando en esta ecuación $\mathbf{A}$ por la FIM:

$$\mathbf{u}^T\mathbf{I}_\lambda\mathbf{u} = \mathbf{u}^T\Big\{E_{\mathbf{x}\sim p_\lambda}\Big[G_\lambda(\mathbf{X})G_\lambda(\mathbf{X})^T\Big]\Big\}\mathbf{u} \qquad ,$$

y usando la propiedad de la linealidad de la esperanza se tiene que:

$$\mathbf{u}^T\Big\{E_{\mathbf{x}\sim p_\lambda}\Big[G_\lambda(\mathbf{X})G_\lambda(\mathbf{X})^T\Big]\Big\}\mathbf{u} = E_{\mathbf{x}\sim p_\lambda}\Big[\mathbf{u}^T G_\lambda(\mathbf{X})G_\lambda(\mathbf{X})^T\mathbf{u}\Big],$$

por lo tanto:

$$\mathbf{u}^T\mathbf{I}_\lambda\mathbf{u} = E_{\mathbf{x}\sim p_\lambda}\Big[\parallel G_\lambda(\mathbf{X})^T\mathbf{u} \parallel^2\Big] \geq 0.$$

Apéndice B

Gradientes del logaritmo de la función de verosimilitud para el cálculo de los FV

Para el cálculo de los FV se necesitan calcular los gradientes del logaritmo de la función de verosimilitud $\nabla_\lambda \log\ p_\lambda(\mathbf{x})$ con respecto a los parámetros λ, como se muestra en la ecuación (2.6). Para esto se define:

$$
\begin{aligned}
p_\lambda(\mathbf{x}) &= \sum_{k=1}^{K} w_k p_k(\mathbf{x}) \\
w_k &= \frac{\exp(\alpha_k)}{\sum_{j=1}^{K}\exp(\alpha_j)} \\
p_k(\mathbf{x}) &= \frac{1}{(2\pi)^{D/2}\boldsymbol{\sigma}_k}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\sigma}_k^{-2}(\mathbf{x}-\boldsymbol{\mu}_k)\right] \\
\forall_k : w_k \geq 0, &\qquad \sum_{k=1}^{K} w_k = 1 \\
\lambda &= \{\alpha_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k, k = 1, \cdots, K\} \\
\gamma_k(\mathbf{x}) &= \frac{w_k p_k(\mathbf{x})}{\sum_{j=1}^{K} w_j p_j(\mathbf{x})},
\end{aligned}
$$

en donde se usa la aproximación diagonal $\boldsymbol{\sigma}_k$ a la matriz de covarianza $\boldsymbol{\Sigma}_k$. Se aclara que en estas ecuaciones las operaciones de producto, división y exponenciación entre vectores se realizan término a término. A continuación se muestran los desarrollos para llegar a los gradientes que aparecen en las ecuaciones (2.14), (2.15) y (2.16).

Gradiente con respecto a α_k

$$
\begin{aligned}
\nabla_{\alpha_k}\log\ p_\lambda(\mathbf{x}) = \nabla_{\alpha_k}\log\sum_{j=1}^{K} w_j p_j(\mathbf{x}) = \nabla_{\alpha_k}\log f(\alpha_k) = \frac{1}{f(\alpha_k)}\nabla_{\alpha_k} f(\alpha_k) \\
= \frac{1}{\sum_{k=1}^{K} w_k p_k(\mathbf{x})}\nabla_{\alpha_k}\left[\sum_{k=1}^{K} w_k p_k(\mathbf{x})\right]
\end{aligned}
$$

$$= \frac{1}{\sum_{k=1}^{K} w_k p_k(\mathbf{x})} \nabla_{\alpha_k} \left[\sum_{k=1}^{K} \frac{\exp(\alpha_k) p_k(\mathbf{x})}{\sum_{j=1}^{K} \exp(\alpha_j)} \right]$$
$$= \frac{1}{\sum_{k=1}^{K} w_k p_k(\mathbf{x})} \nabla_{\alpha_k} \left[\frac{1}{\sum_{j=1}^{K} \exp(\alpha_j)} \sum_{j=1}^{K} \exp(\alpha_j) p_j(\mathbf{x}) \right]$$

desarrollando la derivada del producto de dos funciones:

$$\nabla_{\alpha_k} \left[\frac{1}{\sum_{j=1}^{K} \exp(\alpha_j)} \sum_{j=1}^{K} \exp(\alpha_j) p_k(\mathbf{x}) \right] = \frac{-\exp(\alpha_k) \left[\sum_{j=1}^{K} \exp(\alpha_j) p_j(\mathbf{x}) \right]}{\left[\sum_{j=1}^{K} \exp(\alpha_j) \right]^2} + \frac{\exp(\alpha_k) p_k(\mathbf{x})}{\sum_{j=1}^{K} \exp(\alpha_j)}$$
$$= -w_k \sum_{j=1}^{K} w_j p_j(\mathbf{x}) + w_k p_k(\mathbf{x})$$

reemplazando este gradiente en el gradiente inicial:

$$\nabla_{\alpha_k} \log\ p_\lambda(\mathbf{x}) = \frac{1}{\sum_{k=1}^{K} w_k p_k(\mathbf{x})} \left[-w_k \sum_{j=1}^{K} w_j p_j(\mathbf{x}) + w_k p_k(\mathbf{x}) \right]$$
$$= -w_k + \frac{w_k p_k(\mathbf{x})}{\sum_{k=1}^{K} w_k p_k(\mathbf{x})}$$

Por último, notando que el último factor son las responsabilidades, se llega a la ecuación (2.14):

$$\nabla_{\alpha_k} \log\ p_\lambda(\mathbf{x}) = -w_k + \gamma_k(\mathbf{x})$$

Gradiente con respecto a $\boldsymbol{\mu}_k$

$$\nabla_{\boldsymbol{\mu}_k} \log\ p_\lambda(\mathbf{x}) = \nabla_{\boldsymbol{\mu}_k} \log \sum_{j=1}^{K} w_j \frac{1}{(2\pi)^{D/2} \boldsymbol{\sigma}_j} \exp\left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\sigma}_j^{-2} (\mathbf{x} - \boldsymbol{\mu}_j) \right]$$
$$= \nabla_{\boldsymbol{\mu}_k} \log f(\boldsymbol{\mu}_k) = \frac{1}{f(\boldsymbol{\mu}_k)} \nabla_{\boldsymbol{\mu}_k} f(\boldsymbol{\mu}_k)$$

$$\nabla_{\boldsymbol{\mu}_k} f(\boldsymbol{\mu}_k) = \nabla_{\boldsymbol{\mu}_k} \left[\sum_{j=1}^{K} w_j \frac{1}{(2\pi)^{D/2} \boldsymbol{\sigma}_j} \exp\left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\sigma}_j^{-2} (\mathbf{x} - \boldsymbol{\mu}_j) \right] \right]$$
$$= \nabla_{\boldsymbol{\mu}_k} \left[w_k \frac{1}{(2\pi)^{D/2} \boldsymbol{\sigma}_k} \exp\left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\sigma}_k^{-2} (\mathbf{x} - \boldsymbol{\mu}_k) \right] \right]$$
$$= w_k \frac{1}{(2\pi)^{D/2} \boldsymbol{\sigma}_k} \exp\left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\sigma}_k^{-2} (\mathbf{x} - \boldsymbol{\mu}_k) \right] \nabla_{\boldsymbol{\mu}_k} \left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\sigma}_k^{-2} (\mathbf{x} - \boldsymbol{\mu}_k) \right]$$
$$= w_k \frac{1}{(2\pi)^{D/2} \boldsymbol{\sigma}_k} \exp\left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\sigma}_k^{-2} (\mathbf{x} - \boldsymbol{\mu}_k) \right] \left(\frac{\mathbf{x} - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k^2} \right)$$

Reemplazando $f(\boldsymbol{\mu}_k)$ y $\nabla_{\boldsymbol{\mu}_k} f(\boldsymbol{\mu}_k)$ en la ecuación del gradiente:

$$\begin{aligned}\nabla_{\boldsymbol{\mu}_k} \log\ p_\lambda(\mathbf{x}) &= \frac{w_k \frac{1}{(2\pi)^{D/2}\boldsymbol{\sigma}_k} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\sigma}_k^{-2}(\mathbf{x}-\boldsymbol{\mu}_k)\right]}{\sum_{j=1}^K w_j \frac{1}{(2\pi)^{D/2}\boldsymbol{\sigma}_j} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T\boldsymbol{\sigma}_j^{-2}(\mathbf{x}-\boldsymbol{\mu}_j)\right]}\left(\frac{\mathbf{x}-\boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k^2}\right)\\ &= \frac{w_k p_k(\mathbf{x})}{\sum_{j=1}^K w_j p_j(\mathbf{x})}\left(\frac{\mathbf{x}-\boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k^2}\right)\end{aligned}$$

Por último, notando que el primer factor de la ecuación anterior son las responsabilidades, se obtiene la ecuación (2.15):

$$\nabla_{\boldsymbol{\mu}_k} \log\ p_\lambda(\mathbf{x}) = \gamma_k(\mathbf{x})\left(\frac{\mathbf{x}-\boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k^2}\right)$$

Gradiente con respecto a $\boldsymbol{\sigma}_k$

$$\begin{aligned}\nabla_{\boldsymbol{\sigma}_k} \log\ p_\lambda(\mathbf{x}) &= \nabla_{\boldsymbol{\sigma}_k} \log \sum_{j=1}^K w_j \frac{1}{(2\pi)^{D/2}\boldsymbol{\sigma}_j} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T\boldsymbol{\sigma}_j^{-2}(\mathbf{x}-\boldsymbol{\mu}_j)\right]\\ &= \nabla_{\boldsymbol{\sigma}_k} \log f(\boldsymbol{\sigma}_k) = \frac{1}{f(\boldsymbol{\sigma}_k)}\nabla_{\boldsymbol{\sigma}_k} f(\boldsymbol{\sigma}_k)\end{aligned}$$

calculando el gradiente de $f(\boldsymbol{\sigma}_k)$:

$$\begin{aligned}\nabla_{\boldsymbol{\sigma}_k} f(\boldsymbol{\sigma}_k) &= \nabla_{\boldsymbol{\sigma}_k}\left[\sum_{j=1}^K w_j \frac{1}{(2\pi)^{D/2}\boldsymbol{\sigma}_j} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T\boldsymbol{\sigma}_j^{-2}(\mathbf{x}-\boldsymbol{\mu}_j)\right]\right]\\ &= \nabla_{\boldsymbol{\sigma}_k}\left[w_k \frac{1}{(2\pi)^{D/2}\boldsymbol{\sigma}_k} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\sigma}_k^{-2}(\mathbf{x}-\boldsymbol{\mu}_k)\right]\right]\\ &= \frac{w_k}{(2\pi)^{D/2}}\nabla_{\boldsymbol{\sigma}_k}\left[\frac{1}{\boldsymbol{\sigma}_k} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\sigma}_k^{-2}(\mathbf{x}-\boldsymbol{\mu}_k)\right]\right]\end{aligned}$$

$$\begin{aligned}\nabla_{\boldsymbol{\sigma}_k} f(\boldsymbol{\sigma}_k) &= \frac{w_k}{(2\pi)^{D/2}}\left\{\nabla_{\boldsymbol{\sigma}_k}\left(\frac{1}{\boldsymbol{\sigma}_k}\right)\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\sigma}_k^{-2}(\mathbf{x}-\boldsymbol{\mu}_k)\right] + \frac{1}{\boldsymbol{\sigma}_k}\nabla_{\boldsymbol{\sigma}_k}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\sigma}_k^{-2}(\mathbf{x}-\boldsymbol{\mu}_k)\right]\right\}\\ &= \frac{w_k}{(2\pi)^{D/2}}\left\{-\frac{1}{\boldsymbol{\sigma}_k^2}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\sigma}_k^{-2}(\mathbf{x}-\boldsymbol{\mu}_k)\right] + \frac{1}{\boldsymbol{\sigma}_k}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\sigma}_k^{-3}(\mathbf{x}-\boldsymbol{\mu}_k)\right]\right\}\\ &= \frac{w_k}{(2\pi)^{D/2}\boldsymbol{\sigma}_k}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\sigma}_k^{-2}(\mathbf{x}-\boldsymbol{\mu}_k)\right]\left[(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\sigma}_k^{-3}(\mathbf{x}-\boldsymbol{\mu}_k) - \boldsymbol{\sigma}_k^{-1}\right]\end{aligned}$$

Reemplazando $f(\boldsymbol{\sigma}_k)$ y $\nabla_{\boldsymbol{\sigma}_k} f(\boldsymbol{\sigma}_k)$ en la ecuación del gradiente:

$$\begin{aligned}\nabla_{\boldsymbol{\sigma}_k} \log\ p_\lambda(\mathbf{x}) &= \frac{w_k \frac{1}{(2\pi)^{D/2}\boldsymbol{\sigma}_k} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\sigma}_k^{-2}(\mathbf{x}-\boldsymbol{\mu}_k)\right]}{\sum_{j=1}^K w_j \frac{1}{(2\pi)^{D/2}\boldsymbol{\sigma}_j} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T\boldsymbol{\sigma}_j^{-2}(\mathbf{x}-\boldsymbol{\mu}_j)\right]}\left[\frac{(\mathbf{x}-\boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^3} - \frac{1}{\boldsymbol{\sigma}_k}\right]\\ &= \frac{w_k p_k(\mathbf{x})}{\sum_{j=1}^K w_j p_j(\mathbf{x})}\left[\frac{(\mathbf{x}-\boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^3} - \frac{1}{\boldsymbol{\sigma}_k}\right]\end{aligned}$$

Nuevamente, notando que el primer factor de la ecuación anterior son las responsabilidades, se obtiene la ecuación (2.16):

$$\nabla_{\boldsymbol{\sigma}_k} \log\ p_\lambda(\mathbf{x}) = \gamma_k(\mathbf{x}_i) \left[\frac{(\mathbf{x}_i - \boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^3} - \frac{1}{\boldsymbol{\sigma}_k} \right]$$

Apéndice C

Distribución de Poisson

La distribución de Poisson es una distribución de probabilidad discreta, que expresa a partir de una frecuencia de ocurrencia media, la probabilidad de que ocurra un determinado número de eventos durante cierto período de tiempo. Concretamente, se especializa en la probabilidad de ocurrencia de sucesos con probabilidades muy pequeñas, o sucesos "raros".

Se dice que una variable aleatoria x tiene una distribución de Poisson con parámetro θ si toma los valores $k = 0, 1, 2, \ldots, \infty$ donde:

$$p_\theta(x = k) \stackrel{\text{def}}{=} \frac{\theta^k \exp(-\theta)}{k!}$$

Una particularidad interesante de esta distribución es que la media y la varianza tienen el mismo valor, θ. En la figura C.1 se muestra la distribución de probabilidad para varios valores del parámetro θ.

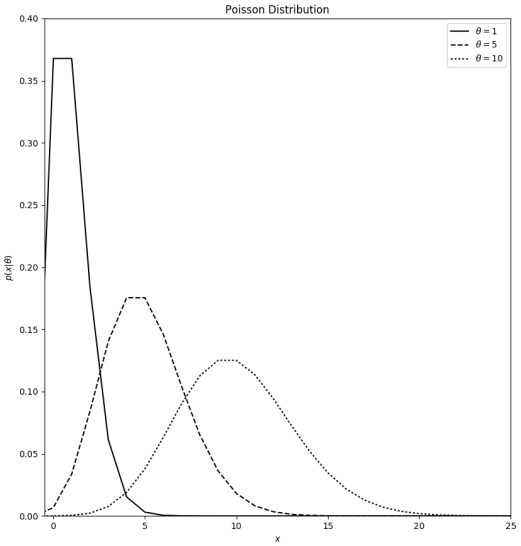


Figura C.1: Función de distribución de probabilidad de Poisson para difirentes valores de θ.

Apéndice D

Gradientes del logaritmo de la función la verosimilitud para el cálculo de los eFV

Para el cálculo de los eFV se necesitan calcular los gradientes del logaritmo de la función de verosimilitud $\nabla_\lambda \log\ p_\lambda(\mathbf{x})$ con respecto a los parámetros λ donde:

$$
\begin{aligned}
p_\lambda(\mathbf{x}) &= \sum_{k=1}^{K} w_k p_k(\mathbf{x}) \\
w_k &= \frac{\exp(\alpha_k)}{\sum_{j=1}^{K}\exp(\alpha_j)} \\
p_k(\mathbf{x}) &= h(\mathbf{x})\exp\left[\langle \boldsymbol{\eta}_k, T_k(\mathbf{x})\rangle - \psi(\boldsymbol{\eta}_k)\right] \\
\forall_k : w_k \geq 0, &\qquad \sum_{k=1}^{K} w_k = 1 \\
\lambda &= \{\alpha_k, \boldsymbol{\eta}_k, k = 1, \cdots, K\} \\
\gamma_k(\mathbf{x}) &= \frac{w_k p_k(\mathbf{x})}{\sum_{j=1}^{K} w_j p_j(\mathbf{x})},
\end{aligned}
$$

Como el cómputo del gradiente de la ecuación (3.14) es similar al cálculo del gradiente de la ecuación (2.14) desarrollado en el apéndice B, solo se muestra el calculo del gradiente con respecto a $\boldsymbol{\eta}_k$ de la ecuación (3.15). Para esto se define lo siguiente:

$$
\begin{aligned}
\nabla_{\boldsymbol{\eta}_k} \log\ p_\lambda(\mathbf{x}) &= \nabla_{\boldsymbol{\eta}_k} \log \sum_{j=1}^{K} w_j h(\mathbf{x}) \exp\left[\langle \boldsymbol{\eta}_j, T_j(\mathbf{x})\rangle - \psi(\boldsymbol{\eta}_j)\right] \\
&= \nabla_{\boldsymbol{\eta}_k} \log f(\boldsymbol{\eta}_k) = \frac{1}{f(\boldsymbol{\eta}_k)} \nabla_{\boldsymbol{\eta}_k} f(\boldsymbol{\eta}_k)
\end{aligned}
$$

$$
\begin{aligned}
\nabla_{\boldsymbol{\eta}_k} f(\boldsymbol{\eta}_k) &= \nabla_{\boldsymbol{\eta}_k} \left\{ \sum_{j=1}^{K} w_j h(\mathbf{x}) \exp\left[\langle \boldsymbol{\eta}_j, T_j(\mathbf{x})\rangle - \psi(\boldsymbol{\eta}_j)\right] \right\} \\
\nabla_{\boldsymbol{\eta}_k} f(\boldsymbol{\eta}_k) &= \nabla_{\boldsymbol{\eta}_k} \{ w_k h(\mathbf{x}) \exp\left[\langle \boldsymbol{\eta}_k, T_k(\mathbf{x})\rangle - \psi(\boldsymbol{\eta}_k)\right] \} \\
\nabla_{\boldsymbol{\eta}_k} f(\boldsymbol{\eta}_k) &= w_k h(\mathbf{x}) \nabla_{\boldsymbol{\eta}_k} \exp\left[\langle \boldsymbol{\eta}_k, T_k(\mathbf{x})\rangle - \psi(\boldsymbol{\eta}_k)\right]
\end{aligned}
$$

$$\nabla_{\boldsymbol{\eta}_k} f(\boldsymbol{\eta}_k) = w_k h(\mathbf{x}) \exp\left[\langle \boldsymbol{\eta}_k, T_k(\mathbf{x})\rangle - \psi(\boldsymbol{\eta}_k)\right] \nabla_{\boldsymbol{\eta}_k}[\langle \boldsymbol{\eta}_k, T_k(\mathbf{x})\rangle - \psi(\boldsymbol{\eta}_k)]$$
$$\nabla_{\boldsymbol{\eta}_k} f(\boldsymbol{\eta}_k) = w_k p_k(\mathbf{x})[T_k(\mathbf{x}) - \nabla\psi(\boldsymbol{\eta}_k)]$$

Reemplazando $f(\boldsymbol{\eta}_k)$ y $\nabla_{\boldsymbol{\eta}_k} f(\boldsymbol{\eta}_k)$ en la ecuación del gradiente completo:

$$\nabla_{\boldsymbol{\eta}_k} \log\ p_\lambda(\mathbf{x}) = \frac{w_k p_k(\mathbf{x})}{\sum_{k=1}^{K} w_k p_k(\mathbf{x})}[T_k(\mathbf{x}) - \nabla\psi(\boldsymbol{\eta}_k)]$$

Por último, notando que el primer factor de la ecuación anterior son las responsabilidades, se obtiene la ecuación (3.15):

$$\nabla_{\boldsymbol{\eta}_k} \log\ p_\lambda(\mathbf{x}) = \gamma_k(\mathbf{x})[T_k(\mathbf{x}) - \nabla\psi(\boldsymbol{\eta}_k)]$$

Apéndice E

Gradiente de $\psi_{(\eta)}$ con respecto η

Para distribuciones que pertenecen a la familia exponencial se cumple que $\nabla_{\boldsymbol{\eta}}\psi(\boldsymbol{\eta}) = \mathbb{E}_{\mathbf{x}\sim p}[T(\mathbf{x})]$, en este apéndice se muestra como se llega a esta igualdad. Sea $p(\mathbf{x})$ una pdf de la familia exponencial definida como en la ecuación (3.7):

$$p(\mathbf{x}) = h(\mathbf{x}) \exp\left[\langle \boldsymbol{\eta}, T(\mathbf{x})\rangle - \psi(\boldsymbol{\eta})\right]$$
$$\int_{\mathcal{X}} p(\mathbf{x})\ d\mathbf{x} = \int_{\mathcal{X}} h(\mathbf{x}) \exp\left[\langle \boldsymbol{\eta}, T(\mathbf{x})\rangle - \psi(\boldsymbol{\eta})\right]\ d\mathbf{x}$$

Usando el hecho de que la integral sobre el espacio muestral es 1 y luego aplicando el gradiente:

$$1 = \int_{\mathcal{X}} h(\mathbf{x}) \exp\left[\langle \boldsymbol{\eta}, T(\mathbf{x})\rangle - \psi(\boldsymbol{\eta})\right]\ d\mathbf{x}$$
$$\nabla_{\boldsymbol{\eta}}(1) = \nabla_{\boldsymbol{\eta}}\left\{\int_{\mathcal{X}} h(\mathbf{x}) \exp\left[\langle \boldsymbol{\eta}, T(\mathbf{x})\rangle - \psi(\boldsymbol{\eta})\right]\ d\mathbf{x}\right\}$$
$$0 = \int_{\mathcal{X}} h(\mathbf{x}) \nabla_{\boldsymbol{\eta}}\{\exp\left[\langle \boldsymbol{\eta}, T(\mathbf{x})\rangle - \psi(\boldsymbol{\eta})\right]\}\ d\mathbf{x}$$
$$0 = \int_{\mathcal{X}} h(\mathbf{x}) \exp\left[\langle \boldsymbol{\eta}, T(\mathbf{x})\rangle - \psi(\boldsymbol{\eta})\right][T(\mathbf{x}) - \nabla_{\boldsymbol{\eta}}\psi(\boldsymbol{\eta})]\ d\mathbf{x}$$

$$\int_{\mathcal{X}} h(\mathbf{x}) \exp\left[\langle \boldsymbol{\eta}, T(\mathbf{x})\rangle - \psi(\boldsymbol{\eta})\right]T(\mathbf{x})\ d\mathbf{x} = \nabla_{\boldsymbol{\eta}}\psi(\boldsymbol{\eta}) \int_{\mathcal{X}} h(\mathbf{x}) \exp\left[\langle \boldsymbol{\eta}, T(\mathbf{x})\rangle - \psi(\boldsymbol{\eta})\right]\ d\mathbf{x}$$
$$\int_{\mathcal{X}} p(\mathbf{x})\ T(\mathbf{x})\ d\mathbf{x} = \nabla_{\boldsymbol{\eta}}\psi(\boldsymbol{\eta}) \int_{\mathcal{X}} p(\mathbf{x})\ d\mathbf{x}$$

Usando la definición de esperanza e integrando nuevamente sobre el espacio muestral del lado derecho de la ecuación de arriba se obtiene:

$$\mathbb{E}_{\mathbf{x}\sim p}[T(\mathbf{x})] = \nabla_{\boldsymbol{\eta}}\psi(\boldsymbol{\eta})$$

Bibliografía

[AJE07] Stian Normann Anfinsen, Robert Jenssen, and Torbjørn Eltoft. Spectral clustering of polarimetric sar data with wishart-derived distance measures. In *Proc. POLinSAR*, volume 7, 2007.

[AN00] Shun-ichi Amari and Hiroshi Nagaoka. Methods of information geometry, volume 191 of translations of mathematical monographs. *American Mathematical Society*, page 13, 2000.

[AOV12] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*, pages 510–517. Ieee, 2012.

[ATC+13] Sandra Avila, Nicolas Thome, Matthieu Cord, Eduardo Valle, and Arnaldo De A AraúJo. Pooling in image representation: The visual codeword point of view. *Computer Vision and Image Understanding*, 117(5):453–465, 2013.

[Bau16] Josef Baumgartner. *Reconocimiento de patrones en campos aleatorios de Markov mediante modelos bayesianos para la agricultura de precisión*. PhD thesis, 2016.

[BBB+10] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[BBH+98] B Brisco, RJ Brown, T Hirose, H McNairn, and K Staenz. Precision agriculture and the role of remote sensing: a review. *Canadian Journal of Remote Sensing*, 24(3):315–327, 1998.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[BMOL+13] Vera Bakic, Sofiene Mouine, Saloua Ouertani-Litayem, Anne Verroust-Blondet, Itheri Yahiaoui, Hervé Goëau, and Alexis Joly. Inria's participation at ImageCLEF 2013 plant identification task. In *CLEF (Online Working Notes/Labs/Workshop) 2013*, 2013.

[BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11), 2001.

[BYM+12] Vera Bakic, Itheri Yahiaoui, Sofiene Mouine, Saloua Litayem Ouertani, Wajih Ouertani, Anne Verroust-Blondet, Hervé Goëau, and Alexis Joly. Inria IMEDIA2's participation at ImageCLEF 2012 plant identification task. In *CLEF (Online Working Notes/Labs/Workshop) 2012*, 2012.

[CAGA14] Carlos Caetano, Sandra Avila, Silvio Guimarães, and Arnaldo de A Araújo. Representing local binary descriptors with bossanova for visual recognition. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pages 49–54. ACM, 2014.

[CAGL14] Qiang Chen, Mani Abedini, Rahil Garnavi, and Xi Liang. IBM research Australia at LifeCLEF 2014: Plant identification task. In *Working notes of CLEF 2014 conference*, 2014.

[CDF+04] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[CHM05] Barbara Caputo, Eric Hayman, and P Mallikarjuna. Class-specific material categorisation. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1597–1604. IEEE, 2005.

[CKF11] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS workshop*, number EPFL-CONF-192376, 2011.

[CLO+12] Michael Calonder, Vincent Lepetit, Mustafa Ozuysal, Tomasz Trzcinski, Christoph Strecha, and Pascal Fua. Brief: Computing a local binary descriptor very fast. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1281–1298, 2012.

[CM14] Archana Chaugule and Suresh N Mali. Evaluation of texture and shape features for classification of four paddy varieties. *Journal of Engineering*, 2014, 2014.

[CMK+14] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3606–3613. IEEE, 2014.

[CPB15] Jyotismita Chaki, Ranjan Parekh, and Samar Bhattacharya. Plant leaf recognition using texture and shape features with neural classifiers. *Pattern Recognition Letters*, 58:61–68, 2015.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[CVS12] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Image categorization using fisher kernels of non-iid image models. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2184–2191. IEEE, 2012.

[CVS13] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Segmentation driven object detection with fisher vectors. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2968–2975. IEEE, 2013.

[DAE08] Anthony P Doulgeris, Stian Normann Anfinsen, and Torbjørn Eltoft. Classification with a non-gaussian model for polsar data. *IEEE Trans. Geosci. Remote Sens.*, 46(10):2999–3009, 2008.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[DE09] Anthony P Doulgeris and Torbjøorn Eltoft. Scale mixture of gaussian modelling of polarimetric sar data. *EURASIP J. Adv. Sig. Process.*, 2010(1):874592, 2009.

[DLR77] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc. Ser. B (Methodological)*, pages 1–38, 1977.

[DM98] Leonardo Dagum and Ramesh Menon. Openmp: an industry standard api for shared-memory programming. *IEEE computational science and engineering*, 5(1):46–55, 1998.

[DOIB12] Andrew Delong, Anton Osokin, Hossam N Isack, and Yuri Boykov. Fast approximate energy minimization with label costs. *Int. J. Computer Vision*, 96(1), 2012.

[DRG+16] Diego Gonzalez Dondo, Javier Andres Redolfi, Martin Griffa, Guillermo Max Steiner, and Luis Rafael Canali. Target tracking system using multiple cameras and bayesian estimation. *IEEE Latin America Transactions*, 14(6):2713–2718, 2016.

[DTY+15] Diego González Dondo, Fernando Trasobares, Leandro Yoaquino, Julián Padilla, and Javier Redolfi. Calibration of multi-camera systems. In *Information Processing and Control (RPIC), 2015 XVI Workshop on*, pages 1–6. IEEE, 2015.

[Esp16] Cecilie Esperbent. Robots: la próxima revolución del campo. *RIA. Revista de investigaciones agropecuarias*, 42(1):8–13, 2016.

[EZW+07] Mark Everingham, Andrew Zisserman, Christopher KI Williams, Luc Van Gool, Moray Allan, Christopher M Bishop, Olivier Chapelle, Navneet Dalal, Thomas Deselaers, Gyuri Dorkó, et al. The pascal visual object classes challenge 2007 (voc2007) results. 2007.

[FCH+08] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[FFC05] Corina C Freitas, Alejandro C Frery, and Antonio H Correia. The polarimetric g distribution for sar data analysis. *Environmetrics*, 16(1):13–31, 2005.

[FHWL14] Masoud Faraki, Mehrtash T Harandi, Arnold Wiliem, and Brian C Lovell. Fisher tensors for classifying human epithelial cells. *Pattern Recognition*, 47(7):2348–2359, 2014.

[FNC11] Alejandro C Frery, Abraão D C Nascimento, and Renato J Cintra. Information theory and image understanding: An application to polarimetric sar imagery. *Chilean J. of Stat.*, 2(2):81–100, 2011.

[FSMST05] Jason Farquhar, Sandor Szedmak, Hongying Meng, and John Shawe-Taylor. Improving "bag-of-keypoints" image categorisation: Generative models and pdf-kernels. 2005.

[GBJ+12] Hervé Goëau, Pierre Bonnet, Alexis Joly, Itheri Yahiaoui, Vera Bakic, Daniel Barthélémy, Nozha Boujemaa, and Jean-François Molino. The ImageCLEF 2012 plant identification task. In *CLEF*, 2012.

[GBJ+13] Hervé Goëau, Pierre Bonnet, Alexis Joly, Vera Bakic, Daniel Barthélémy, Nozha Boujemaa, and Jean-François Molino. The ImageCLEF 2013 plant identification task. In *CLEF*, 2013.

[GDDM14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[GG87] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. In *Readings in Computer Vision*, pages 564–584. Elsevier, 1987.

[GGAM14] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European Conference on Computer Vision*, pages 345–360. Springer, 2014.

[Gir15] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[GJB+14] Hervé Goëau, Alexis Joly, Pierre Bonnet, Souheil Selmi, Jean-François Molino, Daniel Barthélémy, and Nozha Boujemaa. LifeCLEF plant identification task 2014. *CLEF2014 Working Notes. Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014*, pages 598–615, 2014.

[GLT11] Dorian Galvez-Lopez and Juan D Tardos. Real-time loop detection with bags of binary words. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 51–58. IEEE, 2011.

[GNVC02] Pablo M Granitto, Hugo D Navone, Pablo F Verdes, and HA Ceccatto. Weed seeds identification by machine vision. *Computers and Electronics in Agriculture*, 33(2):91–103, 2002.

[Goo63] NR Goodman. Statistical analysis based on a certain multivariate complex gaussian distribution (an introduction). *Ann. Math. Statist.*, 34(1), 1963.

[GVC05] Pablo M Granitto, Pablo F Verdes, and H Alejandro Ceccatto. Large-scale investigation of weed seed identification by machine vision. *Computers and Electronics in Agriculture*, 47(1):15–24, 2005.

[GYM14] Wei Gao, Jian Yang, and Wenting Ma. Land cover classification for polarimetric sar images based on mixture models. *Remote Sensing*, 6(5):3770–3790, 2014.

[Har54] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[HDF12] Jared Heinly, Enrique Dunn, and Jan-Michael Frahm. Comparative evaluation of binary features. In *Computer Vision–ECCV 2012*, pages 759–773. Springer, 2012.

[HHH+15] Phan Thi Thu Hong, Tran Thi Thanh Hai, Vo Ta Hoang, Vu Hai, Thuy Thi Nguyen, et al. Comparative study on vision based rice seed varieties identification. In *Knowledge and Systems Engineering (KSE), 2015 Seventh International Conference on*, pages 377–382. IEEE, 2015.

[HKCL14] Jou-Ken Hsiao, Li-Wei Kang, Ching-Long Chang, and Chih-Yang Lin. Comparative study of leaf image recognition with a novel learning-based approach. In *Science and Information Conference (SAI), 2014*, pages 389–393. IEEE, 2014.

[HKJ16] B. Hou, H. Kou, and L. Jiao. Classification of polarimetric sar images using multilayer autoencoders and superpixels. *IEEE J. of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(7):3072–3081, July 2016.

[HKS11] Elad Hazan, Tomer Koren, and Nati Srebro. Beating sgd: Learning svms in sublinear time. In *Advances in Neural Information Processing Systems*, pages 1233–1241, 2011.

[HO14] Sebastian Haug and Jörn Ostermann. A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks. In *ECCV Workshops (4)*, pages 105–116, 2014.

[HW68] David H Hubel and Torsten N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.

[HWWT14] Yongzhen Huang, Zifeng Wu, Liang Wang, and Tieniu Tan. Feature coding in image classification: A comprehensive study. *IEEE TPAMI*, 36(3):493–506, 2014.

[JDS09] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. On the burstiness of visual elements. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1169–1176. IEEE, 2009.

[JDSP10] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE, 2010.

[JGHV04] Alfons Juan, José García-Hernández, and Enrique Vidal. Em initialisation for bernoulli mixture learning. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 635–643. Springer, 2004.

[JH+99] Tommi S Jaakkola, David Haussler, et al. Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems*, pages 487–493, 1999.

[JL16] Licheng Jiao and Fang Liu. Wishart deep stacking network for fast polsar image classification. *IEEE Trans. on Image Processing*, 25(7):3273–3286, 2016.

[JSD+14] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[K+17] Anthony King et al. The future of agriculture. *Nature*, 544(7651):S21–S23, 2017.

[Kad14] Abdul Kadir. A Model of Plant Identification System Using GLCM, Lacunarity And Shen Features. *arXiv preprint arXiv:1410.0969*, 2014.

[KCZ11] Andrea Vedaldi Ken Chatfield, Victor Lempitsky and Andrew Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *Proc. BMVC*, pages 76.1–76.12, 2011.

[KLSS17] Nataliia Kussul, Mykola Lavreniuk, Sergii Skakun, and Andrii Shelestov. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782, 2017.

[KNSS11] A Kadir, LE Nugroho, A Susanto, and PI Santosa. Neural Network Application on Foliage Plant Identification. *International Journal of Computer Applications*, 29(9):15–22, 2011.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[KT15] Cem Kalyoncu and Önsen Toygar. Geometric leaf classification. *Computer Vision and Image Understanding*, 133:102–109, 2015.

[KVJ11] Josip Krapac, Jakob Verbeek, and Frédéric Jurie. Modeling spatial layout with fisher vectors for image categorization. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1487–1494. IEEE, 2011.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[LCDH+90] B Boser Le Cun, John S Denker, D Henderson, Richard E Howard, W Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*. Citeseer, 1990.

[LGA+99] Jong-Sen Lee, Mitchell R Grunes, Thomas L Ainsworth, Li-Jen Du, Dale L Schuler, and Shane R Cloude. Unsupervised classification using polarimetric decomposition and the complex wishart classifier. *IEEE Trans. Geosc. Remote Sens.*, 37(5), 1999.

[LJS99] X Luo, DS Jayas, and SJ Symons. Identification of damaged kernels in wheat using a colour machine vision system. *Journal of cereal science*, 30(1):49–59, 1999.

[Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[LP09] Jong-Sen Lee and Eric Pottier. *Polarimetric radar imaging: from basics to applications*. CRC press, 2009.

[LSP06] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[M+67] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[MAK13] Ricardo José M Melchiori, Susana M Albarenque, and Alejandra Cecilia Kemerer. Uso, adopción y limitaciones de la agricultura de precisión en argentina. *XII^{o} Curso Internacional de Agricultura de Precisión*, 12(2013):07, 2013.

[MIB97] M Susan Moran, Yoshio Inoue, and EM Barnes. Opportunities and limitations for image-based remote sensing in precision crop management. *Remote sensing of Environment*, 61(3):319–346, 1997.

[Min00] Thomas Minka. Estimating a Dirichlet distribution, 2000.

[MJ99] S Majumdar and DS Jayas. Classification of bulk samples of cereal grains using machine vision. *Journal of Agricultural Engineering Research*, 73(1):35–47, 1999.

[MSJ14] Bingpeng Ma, Yu Su, and Frédéric Jurie. Covariance descriptor based on bio-inspired features for person re-identification and face verification. *Image and Vision Computing*, 32(6):379–390, 2014.

[Mur12] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[Nak13] Hideki Nakayama. Nlab-utokyo at ImageCLEF 2013 plant identification task. In *Working notes of CLEF 2013 conference*, 2013.

[OPM02] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE TPAMI*, 24(7):971–987, 2002.

[OVS14] Dan Oneata, Jakob Verbeek, and Cordelia Schmid. Efficient action localization with approximately normalized fisher vectors. In *CVPR 2014-IEEE Conference on Computer Vision & Pattern Recognition*, pages 2545–2552. IEEE, 2014.

[PD07] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[PDCB06] Florent Perronnin, Christopher Dance, Gabriela Csurka, and Marco Bressan. Adapted vocabularies for generic visual categorization. In *European Conference on Computer Vision*, pages 464–475. Springer, 2006.

[Pen06] Xavier Pennec. Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127, 2006.

[PHG12] Sébastien Paris, Xanadu Halkias, and Hervé Glotin. Participation of LSIS/DYNI to ImageCLEF 2012 plant images classification task. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[PKS+13] Naveen Pandey, Satyanarayan Krishna, Shanu Sharma, et al. Automatic seed classification by shape and color features using machine vision technology. *International Journal of Computer Applications Technology and Research*, 2(2):208–213, 2013.

[PLSP10] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3384–3391. IEEE, 2010.

[Pot52] Renfrey Burnard Potts. Some generalized order-disorder transformations. In *Mathematical Proc. of the Cambridge Philosophical Society*, volume 48. Cambridge Univ Press, 1952.

[PP13] Alireza Pazoki and Zohreh Pazoki. Classification system for rain fed wheat grain cultivars using artificial neural network. *African Journal of Biotechnology*, 10(41):8031–8038, 2013.

[PPAFS12] Alireza Pourreza, Hamidreza Pourreza, Mohammad-Hossein Abbaspour-Fard, and Hassan Sadrnia. Identification of nine iranian wheat seed varieties by textural analysis with image processing. *Computers and electronics in agriculture*, 83:102–108, 2012.

[PSM10] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. *Computer Vision–ECCV 2010*, pages 143–156, 2010.

[QPW+17] WS Qureshi, A Payne, KB Walsh, R Linker, O Cohen, and MN Dailey. Machine vision for counting fruit on mango tree canopies. *Precision Agriculture*, 18(2):224–244, 2017.

[RASC14] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN Features off-the-shelf: an Astounding Baseline for Recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE, 2014.

[RBD+09] Arno Ruckelshausen, Peter Biber, Michael Dorna, Holger Gremmes, Ralph Klose, Andreas Linz, Florian Rahe, Rainer Resch, Marius Thiel, Dieter Trautz, et al. Bonirob–an autonomous field robot platform for individual plant phenotyping. *Precision agriculture*, 9(841):1, 2009.

[RDS+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[RGDPC16] Javier A Redolfi, Diego González Dondo, Julián Antonio Pucheta, and Luis R Canali. Clasificación de variedades de semillas de trigo usando visión por computadora. In *VIII Congreso Argentino de AgroInformática (CAI-2016)-JAIIO 45 (Tres de Febrero, 2016).*, 2016.

[RMWF13] Deepak K Ray, Nathaniel D Mueller, Paul C West, and Jonathan A Foley. Yield trends are insufficient to double global crop production by 2050. *PloS one*, 8(6):e66428, 2013.

[RR16] Cristian Rodríguez Rivero. *Modelos no lineales de pronóstico de series temporales basados en inteligencia computacional para soporte en la toma de decisiones agrícolas*. PhD thesis, 2016.

[RS12] Javier Redolfi and Jorge Sánchez. Leveraging robust signatures for mobile robot semantic localization. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012. RobotVision@ImageCLEF Best Paper Award.

[RSF17] Javier Redolfi, Jorge Sánchez, and Ana Georgina Flesia. Fisher vectors for polsar image classification. *IEEE Geoscience and Remote Sensing Letters*, 14(11):2057–2061, 2017.

[RSJW15] Lankapalli Ravikanth, Chandra B Singh, Digvir S Jayas, and Noel DG White. Classification of contaminants from wheat using near-infrared hyperspectral imaging. *Biosystems Engineering*, 135:73–86, 2015.

[RSP15a] Javier A Redolfi, Jorge A Sánchez, and Julián A Pucheta. Fisher vectors for leaf image classification: An experimental evaluation. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 298–305. Springer International Publishing, 2015.

[RSP15b] Javier A Redolfi, Jorge A Sánchez, and Julián A Pucheta. Identificación de hojas de plantas usando vectores de fisher. *Proceedings of ASAI 2015 Argentine Symposium on Artificial Intelligence*, pages 80–87, 2015.

[RW84] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26:195–239, 1984.

[SAA+16] Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, and Darko Stefanovic. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational intelligence and neuroscience*, 2016, 2016.

[Sán11] Jorge Sánchez. *Modelos eficientes para la clasificación de imágenes en gran escala*. PhD thesis, 2011.

[SBG14] Asma Rejeb Sfar, Nozha Boujemaa, and Donald Geman. Confidence Sets for Fine-Grained Categorization and Plant Species Identification. *International Journal of Computer Vision*, pages 1–21, 2014.

[SEZ+14] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2014.

[SGSS16] Arti Singh, Baskar Ganapathysubramanian, Asheesh Kumar Singh, and Soumik Sarkar. Machine learning for high-throughput stress phenotyping in plants. *Trends in plant science*, 21(2):110–124, 2016.

[SJN13] Christophe Saint-Jean and Frank Nielsen. A new implementation of k-MLE for mixture modeling of wishart distributions. In *Geom. Sci. of Inf.*, pages 249–256. Springer, 2013.

[SK10] Jason Sanders and Edward Kandrot. *CUDA by example: an introduction to general-purpose GPU programming*. Addison-Wesley Professional, 2010.

[SLCS14] Neda Salamati, Diane Larlus, Gabriela Csurka, and Sabine Süsstrunk. Incorporating near-infrared information into semantic image segmentation. *CoRR*, 2014.

[SLJ+] C Szegedy, W Liu, Y Jia, P Sermanet, S Reed, and D Anguelov. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

[SPMV13] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013.

[SR15] Jorge Sánchez and Javier Redolfi. Exponential family Fisher vector for image classification. *Pattern Recognition Letters*, 59:26–32, 2015.

[Sra11] Suvrit Sra. Positive definite matrices and the symmetric Stein divergence. *preprint*, 2011.

[SuHJ12] Gaurav Sharma, Sibt ul Hussain, and Frédéric Jurie. Local higher-order statistics (lhs) for texture categorization and facial analysis. *Computer Vision–ECCV 2012*, pages 1–12, 2012.

[SVV15] Fernando Miguel Scaramuzza, Juan Pablo Vélez, and Diego Daniel Villarroel. Adopción de la agricultura de precisión en argentina, evolución en los principales segmentos. *XV^o Curso Internacional de Agricultura de Precisión*, 15(2015):06, 2015.

[SZ03] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1470–1477 vol.2, Oct 2003.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[TBP+08] Alberto Tellaeche, Xavier P BurgosArtizzu, Gonzalo Pajares, Angela Ribeiro, and César Fernández-Quintanilla. A new vision-based approach to differential spraying in precision agriculture. *computers and electronics in agriculture*, 60(2):144–155, 2008.

[TE11] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1521–1528. IEEE, 2011.

[TLPG14] Hedi Tabia, Hamid Laga, David Picard, and Philippe-Henri Gosselin. Covariance descriptors for 3d shape matching and retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4185–4192, 2014.

[TPM06] Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. In *Computer Vision–ECCV 2006*, pages 589–600. Springer, 2006.

[TS15] Lina Tang and Guofan Shao. Drone remote sensing for forestry research and practices. *Journal of Forestry Research*, 26(4):791–797, 2015.

[USS16] Yusuke Uchida, Shigeyuki Sakazawa, and Shin'ichi Satoh. Image retrieval with fisher vectors of binary features. *ITE Transactions on Media Technology and Applications*, 4(4):326–336, 2016.

[VGVSG10] Jan C Van Gemert, Cor J Veenman, Arnold WM Smeulders, and Jan-Mark Geusebroek. Visual word ambiguity. *IEEE transactions on pattern analysis and machine intelligence*, 32(7):1271–1283, 2010.

[VR14] Kanesh Venugoban and Amirthalingam Ramanan. Image classification of paddy field insect pests using gradient-based features. *International Journal of Machine Learning and Computing*, 4(1):1, 2014.

[WBX+07] Stephen Gang Wu, Forrest Sheng Bao, Eric You Xu, Yu-Xuan Wang, Yi-Fan Chang, and Qiao-Liang Xiang. A leaf recognition algorithm for plant classification using probabilistic neural network. In *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, pages 11–16. IEEE, 2007.

[WCM05] John Winn, Antonio Criminisi, and Thomas Minka. Object categorization by learned universal visual dictionary. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1800–1807. IEEE, 2005.

[WDL99] D Wang, FE Dowell, and RE Lacey. Single wheat kernel color classification by using near-infrared reflectance spectra. *Cereal chemistry*, 76(1):30–33, 1999.

[WLW+14] Anran Wang, Jiwen Lu, Gang Wang, Jianfei Cai, and Tat-Jen Cham. Multi-modal unsupervised feature learning for RGB-D scene labeling. In *ECCV*, 2014.

[WSM+14] Zhaobin Wang, Xiaoguang Sun, Yide Ma, Hongjuan Zhang, Yurun Ma, Weiying Xie, and Yaonan Zhang. Plant recognition based on intersecting cortical model. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pages 975–980. IEEE, 2014.

[WZT+16] Hanning Wang, Zhimin Zhou, John Turnbull, Qian Song, and Feng Qi. Pol-SAR classification based on generalized polar decomposition of mueller matrix. *IEEE Geosc. Remote Sens. Lett.*, 13(4):565–569, 2016.

[YAT12] Berrin A Yanikoglu, Erchan Aptoula, and Caglar Tirkaz. Sabanci-Okan system at ImageClef 2012: Combining features and classifiers for plant identification. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[YAY13] Berrin Yanikoglu, Erchan Aptoula, and S Tolga Yildiran. Sabanci-Okan system at ImageCLEF 2013 plant identification competition. In *Working notes of CLEF 2013 conference*, 2013.

[ZDGD14] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[ZF14] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[ZMZ16] Lu Zhang, Wenping Ma, and Dan Zhang. Stacked sparse autoencoder in polsar data classification using local spatial information. *IEEE Geosc. Remote Sens. Lett.*, 13(9):1359–1363, Sept 2016.

[ZSZM15] Lamei Zhang, Liangjie Sun, Bin Zou, and Wooil M Moon. Fully polarimetric sar image classification via sparse representation and polarimetric features. *IEEE J. Sel. Topics Appl. Earth Observ.*, 8(8):3923–3932, 2015.

[ZWN+09] Yudong Zhang, Lenan Wu, Nabil Neggaz, Shuihua Wang, and Geng Wei. Remote-sensing image classification based on an improved probabilistic neural network. *Sensors*, 9(9):7516–7539, 2009.

[ZYZH10] Xi Zhou, Kai Yu, Tong Zhang, and Thomas Huang. Image classification using super-vector coding of local image descriptors. *Computer Vision–ECCV 2010*, pages 141–154, 2010.

[ZZBC13] Yu Zhang, Chao Zhu, Stéphane Bres, and Liming Chen. Encoding local binary descriptors by bag-of-features with hamming distance for visual object categorization. In *ECIR*, 2013.