TESIS DE DOCTORADO

DEEP LEARNING BASED CLASSIFICATION TECHNIQUES FOR HYPERSPECTRAL IMAGES IN REAL TIME

Presentada por:

Jorge Alberto Suárez Garea

Dirigida por:

Francisco Santiago Argüello Pedreira

Dora Blanco Heras

ESCUELA DE DOCTORADO INTERNACIONAL DE LA UNIVERSIDAD DE SANTIAGO DE COMPOSTELA

PROGRAMA DE DOCTORADO EN INVESTIGACIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

SANTIAGO DE COMPOSTELA

2021

USC
UNIVERSIDADE
DE SANTIAGO
DE COMPOSTELA

DECLARACIÓN DEL AUTOR DE LA TESIS

Deep Learning Based Classification Techniques for Hyperspectral Images in Real Time

Don Jorge Alberto Suárez Garea

Presento mi tesis, siguiendo el procedimiento adecuado al Reglamento, y declaro que:

1. *La tesis abarca los resultados de la elaboración de mi trabajo.*
2. *En su caso, en la tesis se hace referencia a las colaboraciones que tuvo este trabajo.*
3. *La tesis es la versión definitiva presentada para su defensa y coincide con la versión enviada en formato electrónico.*
4. *Confirmo que la tesis no incurre en ningún tipo de plagio de otros autores ni de trabajos presentados por mí para la obtención de otros títulos.*

En Santiago de Compostela, 29 de abril de 2021

Fdo. Jorge Alberto Suárez Garea

USC
UNIVERSIDADE
DE SANTIAGO
DE COMPOSTELA

AUTORIZACIÓN DEL DIRECTOR/TUTOR DE LA TESIS

Deep Learning Based Classification Techniques for Hyperspectral Images in Real Time

Don Francisco Santiago Argüello Pedreira, Profesor Titular de Universidad del Área de Arquitectura de Computadores de la Universidad de Santiago de Compostela

Doña Dora Blanco Heras, Profesora Titular de Universidad del Área de Arquitectura de Computadores de la Universidad de Santiago de Compostela

INFORMAN:

*Que la presente tesis, se corresponde con el trabajo realizado por **Don Jorge Alberto Suárez Garea**, bajo nuestra dirección/tutorización, y autorizamos su presentación, considerando que reúne los requisitos exigidos en el Reglamento de Estudios de Doctorado de la USC, y que como directores/tutores de esta no incurre en las causas de abstención establecidas en la Ley 40/2015.*

De acuerdo con el artículo 41 del Reglamento de Estudios de Doctorado, declaramos también que la presente tesis doctoral es idónea para ser defendida en base a la modalidad de COMPENDIO DE PUBLICACIONES, en los que la participación del doctorando/a fue decisiva para su elaboración y las publicaciones se ajustan al Plan de Investigación.

En Santiago de Compostela, 29 de abril de 2021

Fdo. Francisco Santiago Argüello Pedreira
Director/a tesis

Fdo. Dora Blanco Heras
Director/a tesis

USC
UNIVERSIDADE
DE SANTIAGO
DE COMPOSTELA

To my daughter Eli

The strongest person I have ever met. I miss you.

USC
UNIVERSIDADE
DE SANTIAGO
DE COMPOSTELA

Nunca dejes de luchar y sonreír.
Todo tiene su recompensa.
Yo puedo con todo.
El proceso es lento pero todo llega.

Eli

USC
UNIVERSIDADE
DE SANTIAGO
DE COMPOSTELA

Acknowledgments

I would like to use these lines to thank all those people who in one way or another have helped in carrying out this work.

First, I would like to thank, sincerely as warmly, my two supervisors. Dora Blanco Heras has been the most important pillar on which this thesis has been built. In many ways, she has behaved more like a friend than a tutor and without her advice and support, both educational and emotional, this project would have been much more difficult to complete. I have to thank Francisco Argüello Pedreira for his guidance, valuable feedback and support. I would also like to acknowledge the support of the Department of Electronics and Computer Science, specially that of the Computer Architecture Group, and the Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS) at the University of Santiago de Compostela, for providing the necessary resources and technical support required by this project. I want to express my appreciation to Diego R. Llanos and Arturo González Escribano, members of the Computer Science Department at the University of Valladolid, for the good treatment they have always given to me.

I would also like to thank all my colleagues at the CiTIUS. Especially Mauro, who has instructed me in some very important aspects of life and Brais, who has always been there to lend a hand, but also: Ahmad, Diego, Rosa, Fernando, María, Andrés, Félix, Anas, Jorge, Feras, Ricardo, Nico, Esteban, Jorge Suárez, Caba, Chema, Antonio Loureiro, Nagy, Manisha, Paulo, and all those who at some point have crossed my path. Also a special thanks to all the members of my group: Pablo, Javier, Álvaro Ordoñez, Álvaro Acción, Pedro and Sergio.

My gratitude also goes to Prof. Lorenzo Bruzzone, University of Trento, and Prof. Begüm Demir, University of Berlin for their advice and kind support during my stay in the RSLAB research group at Trento. In this sense, I would also like to include all the people I met

during that visit, especially Ana María, Adamo, Mahdi, Akshara, Sanchari, Massimo Santoni, Francesca, Tatiana, and Christopher.

Also, my deepest thanks to my family and close friends for their endless patience and encouragement in difficult moments during all these years.

I could not end this section without thanking with all my heart a person who taught me that I should never give up, even if sometimes I find it difficult to follow her teachings. She knows who I mean.

Finally, I would like to thank the institutions that provided funding for the development of this thesis: Consellería de Cultura, Educación e Ordenación Universitaria (accreditation 2016-2019, ED431G/08, GRC2014/008, and ED431C 2018/19); Ministry of Science and Innovation, Government of Spain, cofounded by the FEDER funds of European Union [TIN2013-41129-P and TIN2016-76373-P]; Civil Program UAVs Initiative, promoted by the Xunta de Galicia and developed in partnership with the Babcock company to promote the use of unmanned technologies in civil services; Junta de Castilla y León - ERDF (PROPHET Project) [grant number VA082P17]; European Regional Development Fund (ERDF); and European Union (European Social Fund - ESF).

29 de abril de 2021

Resumen

El propósito de esta tesis es el desarrollo de esquemas eficientes, basados en el uso de redes neuronales de aprendizaje profundo, para la clasificación de imágenes multi e hiperespectrales de cobertura terrestre obtenidas mediante sensado remoto. Por esquemas eficientes se entienden aquellos que son capaces de obtener buenos resultados en términos de precisión de la clasificación y que, además, puedan ser computados en tiempo reducido. Respecto de las plataformas computacionales, se considerarán arquitecturas multinúcleo y unidades de procesamiento gráfico (*GPUs, Graphics Processing Units*).

El sensado remoto puede definirse como la adquisición de información de una determinada escena sin entrar en contacto físico con ella, mediante el uso de sensores, principalmente ubicados en plataformas aéreas, que capturan información en diferentes rangos del espectro electromagnético. En función del sensor utilizado para la captura de datos, los valores de reflectancia capturados pueden ir desde el infrarrojo cercano hasta el ultravioleta, incluyendo también el espectro visible. El conjunto de datos adquirido está definido por el tamaño del rango espectral capturado, así, si el rango incluye únicamente unas pocas frecuencias discretas, el conjunto de datos se denomina imagen multiespectral, mientras que si el rango es más amplio y normalmente continuo, del orden de cien bandas o más, el conjunto de datos pasará a denominarse imagen hiperespectral. Esto también influye en las técnicas que se deben seleccionar para el tratamiento de los datos, ya que un mayor número de bandas implica una mayor cantidad de información para ser procesada y, por lo tanto, más potencial de extracción de información relevante a la vez que una complejidad computacional mayor.

En la naturaleza, cada uno de los distintos materiales que podemos observar posee una firma espectral, es decir, un conjunto de valores de reflectancia únicos que los caracteriza. Las características especiales de las imágenes hiperespectrales, que proporcionan información detallada de cada uno de los píxeles que la componen, permiten la distinción entre materiales

físicos y objetos incluso a nivel de pixel. Esta distinción será más precisa cuanto mayor sea el número de bandas de la imagen capturada. La alta dimensionalidad de los datos en este tipo de imágenes presenta nuevos desafíos en técnicas como la separación de información espectral (*unmixing*) [1, 2], la detección de objetivos y anomalías [3, 4], la extracción de características [5, 6] o la caracterización y clasificación de elementos sobre la superficie terrestre [5, 7].

La clasificación de imágenes de la superficie terrestre es uno de los procesos más comunes. Durante ese proceso, cada uno de los píxeles que componen la imagen es clasificado, o sea, le es asignada una etiqueta de clase, acorde a un conjunto de clases previamente definido. Esta clasificación puede ser supervisada o no supervisada. Durante la clasificación no supervisada el algoritmo trabaja sin más información que la que es capaz de extraer del propio conjunto de datos. Por otro lado, la clasificación supervisada necesita de cierta información de referencia, es decir, necesita conocer a que clase pertenecen algunos de los píxeles de la imagen. Esa información será utilizada durante un proceso de aprendizaje que permitirá al algoritmo adquirir el conocimiento necesario para clasificar el resto de la imagen.

Los algoritmos de clasificación basados en aprendizaje automático (*ML, Machine Learning*) tienen un papel muy importante dentro de la literatura referente al sensado remoto [8]. ML engloba a un conjunto de algoritmos que capturan la dinámica del sistema y no requieren de la intervención humana para realizar ciertos cambios. Esto los hace más robustos y menos dependientes de expertos humanos. Por su parte, el aprendizaje profundo (*DL, Deep Learning*) define a un subconjunto de algoritmos del ML, los cuales aprenden a representar el problema como una jerarquía anidada de conceptos donde los conceptos más generales son definidos en relación a conceptos más simples y las representaciones más abstractas son obtenidas en base a otras que lo son menos.

Arquitecturas basadas en ML como las redes neuronales artificiales (*ANNs, Artificial Neural Networks*) [9, 10] o arquitecturas basadas en DL como las redes neuronales artificiales profundas (*DNNs*) o las redes neuronales convolucionales (*CNNs, Convolutional Neural Networks*) [11, 12, 13] han sido utilizadas con éxito para la clasificación de imágenes de sensado remoto. El método estándar de aprendizaje para este tipo de redes consiste en un proceso iterativo, el cual se repite hasta alcanzar un determinado número de iteraciones o hasta conseguir reducir por debajo de un determinado umbral el error producido por la red, es decir, la diferencia entre la salida obtenida por la red y la salida deseada. Dependiendo de la configuración de la red este proceso de aprendizaje puede llegar a ser muy costoso.

Schmidt et al. [14] propusieron un método basado en pesos aleatorios como alternativa al aprendizaje estándar hasta aquel momento, lo que reducía el coste computacional. Esta idea fue aplicada posteriormente por Huang et al. [15] para desarrollar un algoritmo denominado máquina de aprendizaje extremo (*ELM, Extreme Learning Machine*). La idea básica consistía en inicializar de manera aleatoria los pesos de las capas ocultas de una red neuronal, de manera que únicamente se necesitase calcular los pesos de la capa de salida. Basada en la solución de mínimos cuadrados de un sistema lineal general y en el cálculo de la inversa de Moore-Penrose, ELM surgía como alternativa al método de aprendizaje estándar basado en la propagación de pesos. Más tarde, Huang et al. [16] propusieron el uso de funciones basadas en núcleos (*kernels*), para reemplazar la asignación aleatoria de pesos. Trataban así de evitar las diferencias entre los resultados que se obtenían al repetir los mismos experimentos, debidas a la aleatoriedad de los pesos. Esta nueva versión de ELM se denominó KELM (*Kernel-based Extreme Learning Machine*).

Las imágenes hiperespectrales contienen una enorme cantidad de información espectral que puede llegar a ser redundante. Para extraer la información más relevante se suelen usar los llamados métodos de extracción de características. Su aplicación da como resultado la reducción de dimensionalidad de la imagen y entre ellos se encuentran métodos como autocodificadores (*AE, Autoencoders*) [17], el análisis de componentes independientes (*ICA, Independent Component Analysis*) [18] o el análisis de componentes principales (*PCA, Principal Component Analysis*) [19]. PCA es uno de los métodos más utilizados en la literatura para este propósito [20] y permite reducir la dimensionalidad de un conjunto de datos al mismo tiempo que preserva tanta variación presente en la misma como el usuario indique.

Además de la información espectral, que se encuentra implícita en la imagen, existe otro tipo de información, la espacial, que puede ser extraída de los conjuntos de datos. Este tipo de información suele referirse al entorno que rodea a cada uno de los píxeles que conforman la imagen. Diferentes métodos como la segmentación [21], que realiza una partición exhaustiva de la imagen en regiones basándose en un criterio de homogeneidad, o métodos basados en estructuras fijas como los perfiles morfológicos (*MPs, Morphological Profiles*) [5] pueden ser utilizados para extraer información espacial. Los MPs extraen la información espacial de los vecinos más cercanos a un determinado pixel mediante operaciones morfológicas denominadas apertura y cierre, al mismo tiempo que preservan las características geométricas de las estructuras en la imagen. Estas operaciones utilizan elementos estructurantes de diferentes tamaños que recorren la imagen para formar un conjunto de MPs denominado perfil mor-

fológico extendido (*EMP, Extended Morphological Profile*) [22, 23]. Este EMP representa la variabilidad a diferentes niveles de escala de las estructuras contenidas en la imagen.

Existen determinados casos, como puede ser la detección de defectos en superficies [24], el análisis de imágenes médicas [25] o el análisis de la vegetación [26] entre otros, en que el uso de información espacial y espectral puede no ser suficiente para discriminar entre elementos. En estos casos, la información de texturas (que no deja de ser espacial) puede ayudar a mejorar la separabilidad entre clases y, por tanto, aumentar la precisión de la clasificación. La textura de una imagen está relacionada con la distribución espacial de los valores de intensidad en la misma y, como tal, contiene información sobre contraste, uniformidad, rugosidad, etc, lo que permite caracterizar las diferentes regiones de interés dentro de una escena. Existe un amplio abanico de algoritmos utilizados en la extracción de información de texturas [27]: modelado de mezcla Gaussiana (*GMM, Gaussian Mixture Modeling*) [28] o k-means [29, 30] son algunos ejemplos. Tanto para GMM como para k-means, el objetivo es el de generar un diccionario de primitivas llamadas textones, que constituyen la textura.

Con el diccionario de textones generado, podemos hacer uso de algoritmos de codificación de características como VLAD (*Vector of Locally Aggregated Descriptors*) [31] para caracterizar como texturas las diferentes regiones, previamente definidas, de una imagen, en base a las características individuales de cada uno de los píxeles de dichas regiones. Una de las técnicas más comunes utilizada para la extracción de regiones uniformes en imágenes es la segmentación basada en superpíxeles [32, 33]. Los superpíxeles proporcionan una representación compacta de las imágenes, lo que permite aplicar los algoritmos de procesamiento a segmentos en lugar de a píxeles individuales, reduciendo así el coste computacional de los algoritmos, al tiempo que se mantiene la calidad de la solución obtenida. En particular, el algoritmo SLIC (*Simple Linear Iterative Clustering*) [34] permite definir superpíxeles de tamaño y regularidad similares a lo largo de toda la imagen.

La resolución de problemas de clasificación mediante métodos supervisados, como los propuestos en esta tesis, está fuertemente ligada a la existencia de información de referencia. En muchos casos, construir esta información es costoso en tiempo y esfuerzo ya que puede requerir datos de zonas de interés de complicado acceso y contrastar información de diferentes fuentes. En el caso de las imágenes de vegetación analizadas en esta tesis doctoral, ha sido necesario viajar junto con expertos de Aguas de Galicia al borde de los ríos, contrastar la información proporcionada por ellos con inventarios de vegetación autonómicos e imágenes obtenidas por herramientas como Google Earth y finalmente realizar un proceso iterativo de

depurado de la información de referencia utilizando herramientas software desarrolladas por nosotros. Además, si a eso le sumamos la gran cantidad de imágenes nuevas que se adquieren cada día, para su uso con aplicaciones relacionadas con cobertura terrestre, nos encontramos ante una tarea en ocasiones intratable.

Por otra parte, si el problema de clasificación que se quiere resolver implica imágenes correspondientes a diferentes áreas geográficas, o capturadas por diferentes sensores o en diferentes periodos de tiempo, la tarea aumenta su complejidad ya que el aprendizaje obtenido para una imagen no es extrapolable de manera directa a otras. Existe además desplazamiento espectral entre imágenes espectrales debido, por ejemplo, a efectos atmosféricos, de temperatura o incluso al propio sensor encargado de capturar los datos, que es necesario tener en cuenta. Ante esta problemática, surge la necesidad de diseñar esquemas de clasificación que sean capaces de aprovechar la escasa información de referencia disponible y extraer conocimiento de la clasificación de unas imágenes para ser aprovechado para la clasificación de imágenes obtenidas, por ejemplo, en otras zonas geográficas.

La transferencia de conocimiento (*TL, Transfer Learning*) engloba a un conjunto de técnicas que se encargan de aplicar el conocimiento adquirido previamente para una o más tareas en un determinado dominio origen, sobre otra relacionada con la tarea inicial, en un dominio destino [35]. A su vez, las técnicas de adaptación del dominio (*DA, Domain Adaptation*), que son un subconjunto del TL, usan información de referencia de imágenes pertenecientes a un dominio origen para tratar de clasificar imágenes pertenecientes a un dominio destino, de las cuales no se tiene información de referencia. Dentro de las técnicas aplicadas para conseguir DA podemos destacar las técnicas de extracción de características [36, 37], las cuales tratan de reducir el desplazamiento entre los dos dominios mediante la búsqueda de una función que pueda mapear los datos a un nuevo espacio que los defina mejor.

El uso de arquitecturas basadas en DL tales como las CNNs, permite encontrar también la función de mapeo necesaria para poder aplicar el conocimiento de un dominio sobre el otro, es decir, realizar DA. Este tipo de redes está formado básicamente por un conjunto de filtros convolucionales anidados que extraen características de forma jerárquica. Tal y como se comentó anteriormente, estas redes neuronales necesitan de un proceso iterativo para aprender los pesos de la red que, dependiendo de la complejidad de la misma, puede llegar a ser muy costoso. Por ello, se han propuesto diferentes alternativas, sobre este tipo de redes neuronales, para reducir ese coste, como [38], a la que hemos prestado especial interés en esta tesis.

En particular, se ha propuesto sustituir el costoso proceso iterativo de aprendizaje de una CNN mediante el cálculo de los filtros de la red convolucional utilizando la técnica llamada análisis de componentes de transferencia (*TCA, Transfer Component Analysis*). TCA es una técnica de extracción de características especialmente diseñada para DA. El objetivo de TCA es acondicionar las distribuciones de datos entre dominios en un nuevo espacio, permitiendo preservar las propiedades de los datos y al mismo tiempo reducir la desviación entre las distribuciones. Aplicando TCA para el cálculo de los filtros de la red CNN se consigue una reducción del coste computacional al tiempo que se reduce el desplazamiento entre dominios, es decir, se adapta la red CNN al problema de DA.

Dada la complejidad de las técnicas de clasificación y la gran cantidad de datos disponibles en las imágenes con las que se ha trabajado, y a la vista de todo lo anterior, nos planteamos en esta tesis un recorrido por diferentes soluciones basadas en DL para la clasificación de imágenes multi e hiperespectrales que apoye la siguiente hipótesis:

> *Es posible diseñar esquemas de clasificación simples para imágenes multi e hiperespectrales basados en DL, que utilicen información espacial y de texturas para lograr buenos resultados en términos de precisión, y que se puedan ejecutar en tiempo real utilizando infraestructuras computacionales de bajo coste.*

El éxito de esquemas eficientes requerirá de un estudio a fondo para encontrar técnicas que mejoren la precisión de la clasificación adaptadas al modelo de computación de este hardware de bajo coste, en nuestro caso CPUs (*Central Processing Units*) multicore y GPUs.

Las contribuciones principales de esta tesis son las siguientes:

1. **Propuesta de un esquema de clasificación espectral-espacial basado en redes neuronales con pesos aleatorios.** KELM-EMP es un esquema basado en extracción de características, fusión de datos y KELM. El esquema consiste en una primera fase de extracción de características donde, tras una reducción de la dimensionalidad de la imagen empleando un algoritmo PCA, se extrae la información espacial presente en la imagen mediante el uso de MPs. La información espectral y la información espacial se fusionan mediante una suma ponderada, permitiendo ajustar la influencia de cada tipo de información en el clasificador. La clasificación final se lleva a cabo mediante un clasificador basado en KELM. La novedad de este esquema es el uso de una red neuronal como

clasificador, cuya fase de entrenamiento estándar ha sido sustituida por la asignación de pesos mediante el uso de una función kernel. Además, se ha añadido una etapa final de regularización espacial que permite reducir el número de píxeles clasificados de forma aislada y aumentar la precisión.

2. **Propuesta de un esquema basado en texturas, para la identificación de especies vegetales sobre imágenes multiespectrales capturadas por vehículos aéreos no tripulados (*UAVs, Unmanned Aerial Vehicles*).** Se trata de un esquema basado en extracción de características y un sistema de clasificación jerárquico formado por un conjunto de máquinas de soporte vectorial (*SVMs, Support Vector Machines*). En este esquema se utilizan, además de la propia información espectral, información espacial y de texturas. La extracción de información espacial se realiza mediante operaciones morfológicas, mientras que para la información de texturas se utiliza *k*-means [29, 30] y VLAD [31].

   Las imágenes obtenidas por UAVs suelen poseer una gran densidad de información, lo que incrementa el tiempo de procesamiento. Tratando de reducir ese tiempo, en este esquema se utiliza el algoritmo de superpíxeles SLIC. Este algoritmo agrupa los píxeles que poseen rasgos similares dentro de una imagen, generando un mapa de segmentación. De esta forma, para la extracción de características, en lugar de trabajar directamente con cada uno de los píxeles de la imagen, se utilizan los centroides (o representantes) de cada uno de los superpíxeles.

   Un clasificador jerárquico compuesto por cuatro SVMs se encarga de la clasificación final. Los dos primeros SVMs separan las clases en grandes grupos mientras que los dos últimos realizan una clasificación más específica. La novedad de este esquema es por un lado la utilización de superpíxeles en combinación con la información espectral-espacial y de texturas y por el otro el uso de un clasificador jerárquico.

3. **Propuesta de esquemas de clasificación basado en CNNs.** En este caso el trabajo consistió en el estudio preliminar de CNNs mediante la comparativa de diferentes posibilidades de implementación. Se utilizaron diferentes librerías para la ejecución de un esquema de clasificación basado en CNNs mediante el framework de DL llamado Caffe [39]. Después de llevar a cabo una reducción de dimensionalidad sobre la imagen a tratar, se extrae de la imagen los patches correspondientes a las muestras de entrenamiento que serán utilizadas por la CNN para aprender los coeficientes de los filtros

convolucionales. La aportación de este análisis era la de conocer hasta que punto el uso de librerías optimizadas permitían reducir el tiempo de ejecución en un esquema basado en CNNs.

4. **Propuesta de un esquema de clasificación basado en CNNs y TCA para DA.** El esquema denominado TCANet que se propone se basa en el uso de una CNN para extraer características comunes de imágenes pertenecientes a dominios diferentes. Dicha CNN no utiliza el procedimiento estándar de entrenamiento, basado en la propagación de pesos, sino que este ha sido sustituido por el algoritmo TCA para calcular los pesos de los dos filtros convolucionales que componen la red del esquema. Al inicio del esquema se realiza un alineamiento condicional de la covarianza de las dos imágenes, pero solo si esto consigue reducir la distancia entre las distribuciones de las dos imágenes.

   La novedad de este esquema es, por tanto, el uso de la técnica TCA para calcular los coeficientes de la red convolucional. Se reduce así el coste de aprendizaje que conlleva el uso de otras técnicas basadas en la propagación de pesos y el uso de gradientes.

5. **Implementación eficiente de esquemas de clasificación en CPUs multinúcleo y GPUs usando OpenMP y CUDA (*Compute Unified Device Architecture*) respectivamente:**

   - Dos esquemas, uno basado en ELM(ELM-EMP) y otro basado en KELM (KELM-EMP) han sido desarrollados en tres diferentes versiones: CPU, OpenMP y GPU. Se han estudiado distintas estrategias para particionar los datos y diferentes configuraciones de bloques con el fin de explotar al máximo la capacidad computacional de la GPU. También se ha hecho uso de librerías optimizadas.
   - En el esquema basado en texturas para clasificación de imágenes multiespectrales de vegetación se han utilizado estrategias de paralelización mediante OpenMP y GPU que han permitido reducir el tiempo de ejecución.
   - El esquema TCANet desarrollado para DA se ha ejecutado sobre Matlab, haciendo uso del conjunto de herramientas de computación paralela disponibles en este software. Esto permite paralelizar la ejecución del esquema sobre cada uno de los píxeles de la imagen.
   - Los esquemas Caffe-HYCNN y cuDNN-HYCNN son dos implementaciones en GPU del esquema HYCNN. Ambos esquemas son ejecutados utilizando el framework de DL Caffe y funciones optimizadas de la librería cuDNN (*CUDA Deep Neural Network*).

6. **Desarrollo de una aplicación de escritorio para el análisis y procesamiento de imágenes de sensado remoto.** Este software, denominado HypeRvieW, es una herramienta de escritorio que permite analizar las características de imágenes de sensado remoto y aplicar algoritmos de procesado sobre las mismas incluyendo registrado o clasificación entre otras tareas. En referencia al análisis, el software es capaz de mostrar las diferentes bandas que forman la imagen así como también de visualizar la firma espectral de cada uno de sus píxeles. Del mismo modo, permite comparar las firmas espectrales de dos píxeles entre sí o con una colección de firmas espectrales pertenecientes a diferentes materiales. El software dispone de las herramientas básicas de visualización de imágenes como son zoom, desplazamiento a una determinada banda, rotación, escalado y recorte de la imagen.

   En cuanto al procesamiento de imágenes, HypeRvieW permite generar esquemas de clasificación supervisada combinando los diferentes módulos presentes en el propio software o mediante la incorporación de nuevos módulos desarrollados por el usuario. Entre los módulos disponibles podemos destacar un algoritmo de reducción de la dimensionalidad basado en PCA; un algoritmo de extracción de características espaciales basado en operaciones morfológicas; algoritmos de segmentación como SLIC, RQS (*Really Quick Shift*) [40] y Watershed [41, 42]; algoritmos de registrado basados en la transformada rápida de Fourier (*FFT, Fast Fourier Transform*) y la transformada de Fourier fraccional multicapa (*MFFT, Multilayer Fractional Fourier Transform*) [43]; o algoritmos de clasificación basados en ELM y SVM. Una importante característica de esta herramienta es que permite al usuario incorporar sus propios desarrollos al software y utilizarlos para generar nuevos esquemas de clasificación.

   Por último, es importante destacar que HypeRvieW nos ha facilitado mostrar a empresas, estudiantes y público en general, de una manera simple, las posibles aplicaciones de la investigación desarrollada por el grupo de investigación especializado en procesado de imagen multi e hiperespectral, liderado por los directores de esta tesis doctoral.

USC
UNIVERSIDADE
DE SANTIAGO
DE COMPOSTELA

Resumo

O propósito desta tese é o desenvolvemento de esquemas eficientes, baseados no uso de redes neuronais de aprendizaxe profunda, para a clasificación de imaxes multi e hiperespectrais de cobertura terrestre obtidas mediante sensado remoto. Por esquemas eficientes enténdense aqueles que son capaces de obter bos resultados en termos de precisión da clasificación e que, ademais, poidan ser computados en tempo reducido. Respecto das plataformas computacionais, consideraranse arquitecturas multinúcleo e unidades de procesamento gráfico (*GPUs, Graphics Processing Units*).

O sensado remoto pode definirse como a adquisición de información dunha determinada escena sen entrar en contacto físico con ela, mediante o uso de sensores, principalmente situados en plataformas aéreas, que capturan información en diferentes rangos do espectro electromagnético. En función do sensor utilizado para a captura de datos, os valores de reflectancia capturados poden ir desde o infravermello próximo ata o ultravioleta, incluíndo tamén o espectro visible. O conxunto de datos adquirido está definido polo tamaño do rango espectral capturado, así, se o rango inclúe unicamente unhas poucas frecuencias discretas, o conxunto de datos denomínase imaxe multiespectral, mentres que se o rango é máis amplo e normalmente continuo, da orde de cen bandas ou máis, o conxunto de datos pasará a denominarse imaxe hiperespectral. Isto tamén influúe nas técnicas que se deben seleccionar para o tratamento dos datos, xa que un maior número de bandas implica unha maior cantidade de información para ser procesada e, por tanto, máis potencial de extracción de información relevante á vez que unha complexidade computacional maior.

Na natureza, cada un dos distintos materiais que podemos observar posúe unha firma espectral, é dicir, un conxunto de valores de reflectancia únicos que os caracteriza. As características especiais das imaxes hiperespectrais, que proporcionan información detallada de cada un dos píxeles que a compoñen, permiten a distinción entre materiais físicos e obxectos

mesmo a nivel de pixel. Esta distinción será máis precisa canto maior sexa o número de bandas da imaxe capturada. A alta dimensionalidade dos datos neste tipo de imaxes presenta novos desafíos en técnicas como a separación de información espectral (*unmixing*) [1, 2], a detección de obxectivos e anomalías [3, 4], a extracción de características [5, 6] ou a caracterización e clasificación de elementos sobre a superficie terrestre [5, 7].

A clasificación de imaxes da superficie terrestre é un dos procesos máis comúns. Durante ese proceso, cada un dos píxeles que compoñen a imaxe é clasificado, ou sexa, élle asignada unha etiqueta de clase, acorde a un conxunto de clases previamente definido. Esta clasificación pode ser supervisada ou non supervisada. Durante a clasificación non supervisada o algoritmo traballa sen máis información que a que é capaz de extraer do propio conxunto de datos. Doutra banda, a clasificación supervisada necesita de certa información de referencia, é dicir, necesita coñecer a que clase pertencen algúns dos píxeles da imaxe. Esa información será utilizada durante un proceso de aprendizaxe que permitirá ao algoritmo adquirir o coñecemento necesario para clasificar o resto da imaxe.

Os algoritmos de clasificación baseados en aprendizaxe automática (*ML, Machine Learning*) teñen un papel moi importante dentro da literatura referente ao sensado remoto [8]. ML engloba a un conxunto de algoritmos que capturan a dinámica do sistema e non requiren da intervención humana para realizar certos cambios. Isto fainos máis robustos e menos dependentes de expertos humanos. Pola súa banda, a aprendizaxe profunda (*DL, Deep Learning*) define a un subconxunto de algoritmos do ML, os cales aprenden a representar o problema como unha xerarquía aniñada de conceptos onde os conceptos máis xerais son definidos en relación a conceptos máis simples e as representacións máis abstractas son obtidas en base a outras que o son menos.

Arquitecturas baseadas en ML como as redes neuronais artificiais (*ANNs, Artificial Neural Networks*) [9, 10] ou arquitecturas baseadas en DL como as redes neuronais artificiais profundas (*DNNs*) ou as redes neuronais convolucionais (*CNNs, Convolutional Neural Networks*) [11, 12, 13] foron utilizadas con éxito para a clasificación de imaxes de sensado remoto. O método estándar de aprendizaxe para este tipo de redes consiste nun proceso iterativo, o cal se repite ata alcanzar un determinado número de iteracións ou ata conseguir reducir por baixo dun determinado limiar o erro producido pola rede, é dicir, a diferenza entre a saída obtida pola rede e a saída desexada. Dependendo da configuración da rede este proceso de aprendizaxe pode chegar a ser moi custoso.

Schmidt et al. [14] propuxeron un método alternativo á aprendizaxe estándar ata aquel momento, baseado en pesos aleatorios que, por tanto, reducía o custo computacional. Esta idea foi aplicada posteriormente por Huang et al. [15] para desenvolver un algoritmo denominado máquina de aprendizaxe extrema (*ELM, Extreme Learning Machine*). A idea básica consistía en inicializar de maneira aleatoria os pesos das capas ocultas dunha rede neuronal, de maneira que unicamente se necesitase calcular os pesos da capa de saída. Baseada na solución de mínimos cadrados dun sistema lineal xeral e no cálculo da inversa de Moore-Penrose, ELM xurdía como alternativa ao método de aprendizaxe estándar baseada na propagación de pesos. Máis tarde, Huang et al. [16] propuxeron o uso de funcións baseadas en núcleos (*kernels*), para substituír a asignación aleatoria de pesos. Trataban así de evitar as diferenzas entre os resultados que se obtiñan ao repetir os mesmos experimentos, debidas á aleatoriedade dos pesos. Esta nova versión de ELM denominouse KELM (*Kernel-based Extreme Learning Machine*).

As imaxes hiperespectrais conteñen unha enorme cantidade de información espectral que pode chegar a ser redundante. Para extraer a información máis relevante adóitanse usar os chamados métodos de extracción de características. A súa aplicación dá como resultado a redución de dimensionalidade da imaxe e entre eles atópanse métodos como auto-codificadores (*AE, Autoencoders*) [17], a análise de compoñentes independentes (*ICA, Independent Component Analysis*) [18] ou a análise de compoñentes principais (*PCA, Principal Component Analysis*) [19]. PCA é un dos métodos máis utilizados na literatura para este propósito [20] e permite reducir a dimensionalidade dun conxunto de datos ao mesmo tempo que preserva tanta variación presente na mesma como o usuario indique.

Ademais da información espectral, que se atopa implícita na imaxe, existe outro tipo de información, a espacial, que pode ser extraída dos conxuntos de datos. Este tipo de información adoita referirse á contorna que rodea a cada un dos píxeles que conforman a imaxe. Diferentes métodos como a segmentación [21], que realiza unha partición exhaustiva da imaxe en rexións baseándose nun criterio de homoxeneidade, ou métodos baseados en estruturas fixas como os perfís morfolóxicos (*MPs, Morphological Profiles*) [5] poden ser utilizados para extraer información espacial. Os MPs extraen a información espacial dos veciños máis próximos a un determinado pixel mediante operacións morfolóxicas denominadas apertura e peche, ao mesmo tempo que preservan as características xeométricas das estruturas na imaxe. Estas operacións utilizan elementos estructurantes de diferentes tamaños que percorren a imaxe para formar un conxunto de MPs denominado perfil morfolóxico estendido (*EMP, Extended Mor-*

phological Profile) [22, 23]. Este EMP representa a variabilidade a diferentes niveis de escala das estruturas contidas na imaxe.

Existen determinados casos, como pode ser a detección de defectos en superficies [24], a análise de imaxes médicas [25] ou a análise da vexetación [26] entre outros, en que o uso de información espacial e espectral pode non ser suficiente para discriminar entre elementos. Nestes casos, a información de texturas (que non deixa de ser espacial) pode axudar a mellorar a separabilidade entre clases e, por tanto, aumentar a precisión da clasificación. A textura dunha imaxe está relacionada coa distribución espacial dos valores de intensidade na mesma e, como tal, contén información sobre contraste, uniformidade, rugosidade, etc, o que permite caracterizar as diferentes rexións de interese dentro dunha escena. Existe un amplo abanico de algoritmos utilizados na extracción de información de texturas [27]: modelado de mestura Gaussiana (*GMM, Gaussian Mixture Modeling*) [28] ou k-means [29, 30] son algúns exemplos. Tanto para GMM como para k-means, o obxectivo é o de xerar un dicionario de primitivas chamadas textones, que constitúen a textura.

Co dicionario de textones xerado, podemos facer uso de algoritmos de codificación de características como VLAD (*Vector of Locally Aggregated Descriptors*) [31] para caracterizar como texturas as diferentes rexións, previamente definidas, dunha imaxe, en base ás características individuais de cada un dos píxeles das devanditas rexións. Unha das técnicas máis comúns utilizada para a extracción de rexións uniformes en imaxes é a segmentación baseada en superpíxeles [32, 33]. Os superpíxeles proporcionan unha representación compacta das imaxes, o que permite aplicar os algoritmos de procesamento a segmentos en lugar de píxeles individuais, reducindo así o custo computacional dos algoritmos, á vez que se mantén a calidade da solución obtida. En particular, o algoritmo SLIC (*Simple Linear Iterative Clustering*) [34] permite definir superpíxeles de tamaño e regularidade similares ao longo de toda a imaxe.

A resolución de problemas de clasificación mediante métodos supervisados, como os propostos nesta tese, está fortemente ligada á existencia de información de referencia. En moitos casos, construír esta información é custoso en tempo e esforzo xa que pode requirir datos de zonas de interese de complicado acceso e contrastar información de diferentes fontes. No caso das imaxes de vexetación analizadas nesta tese de doutoramento, foi necesario viaxar xunto con expertos de Augas de Galicia ao bordo dos ríos, contrastar a información proporcionada por eles con inventarios de vexetación autonómicos e imaxes obtidas por ferramentas como Google Earth e, finalmente, realizar un proceso iterativo de depurado da información de referencia utilizando ferramentas software desenvolvidas por nós. Ademais, se a iso sumámoslle

a gran cantidade de imaxes novas que se adquiren cada día, para o seu uso con aplicacións relacionadas coa cobertura terrestre, atopámonos ante unha tarefa en ocasións intratable.

Por outra banda, se o problema de clasificación que se quere resolver implica imaxes correspondentes a diferentes áreas xeográficas, ou capturadas por diferentes sensores ou en diferentes períodos de tempo, a tarefa aumenta a súa complexidade, xa que a aprendizaxe obtida para unha imaxe non é extrapolable de maneira directa a outras. Existe ademais desprazamento espectral entre imaxes espectrais debido, por exemplo, a efectos atmosféricos, de temperatura ou mesmo ao propio sensor encargado de capturar os datos que é necesario ter en conta. Ante esta problemática, xorde a necesidade de deseñar esquemas de clasificación que sexan capaces de aproveitar a escasa información de referencia dispoñible e extraer coñecemento da clasificación dunhas imaxes para ser aproveitado para a clasificación de imaxes obtidas, por exemplo, noutras zonas xeográficas.

A transferencia de coñecemento (*TL, Transfer Learning*) engloba un conxunto de técnicas que se encargan de aplicar o coñecemento adquirido previamente para unha ou máis tarefas nun determinado dominio orixe, sobre outra relacionada coa tarefa inicial, nun dominio destino [35]. Á súa vez, as técnicas de adaptación do dominio (*DA, Domain Adaptation*), que son un subconxunto do TL, usan información de referencia de imaxes pertencentes a un dominio orixe para tratar de clasificar imaxes pertencentes a un dominio destino, das cales non se ten información de referencia. Dentro das técnicas aplicadas para conseguir DA, podemos destacar as técnicas de extracción de características [36, 37], as cales tratan de reducir o desprazamento entre os dous dominios mediante a procura dunha función que poida mapear os datos a un novo espazo que os defina mellor.

O uso de arquitecturas baseadas en DL tales como as CNNs, permite atopar tamén a función de mapeo necesaria para poder aplicar o coñecemento dun dominio sobre o outro, é dicir, realizar DA. Este tipo de redes está formado basicamente por un conxunto de filtros convolucionais aniñados que extraen características de forma xerárquica. Tal e como se comentou anteriormente, estas redes neuronais necesitan dun proceso iterativo para aprender os pesos da rede que, dependendo da complexidade da mesma, pode chegar a ser moi custoso. Por iso, propuxéronse diferentes alternativas, sobre este tipo de redes neuronais, para reducir ese custo, como [38], á que prestamos especial interese nesta tese.

En particular, propúxose substituír o custoso proceso iterativo de aprendizaxe dunha CNN mediante o cálculo dos filtros da rede convolucional utilizando a técnica chamada análise de compoñentes de transferencia (*TCA, Transfer Component Analysis*). TCA é unha técnica

de extracción de características especialmente deseñada para DA. O obxectivo de TCA é acondicionar as distribucións de datos entre dominios nun novo espazo, permitindo preservar as propiedades dos datos e ao mesmo tempo reducir a desviación entre as distribucións. Aplicando TCA para o cálculo dos filtros da rede CNN conséguese unha redución do custo computacional á vez que se reduce o desprazamento entre dominios, é dicir, adáptase a rede CNN ao problema de DA.

Dada a complexidade das técnicas de clasificación e a gran cantidade de datos dispoñibles nas imaxes coas que se traballou, e á vista de todo o anterior, plantexámonos nesta tese un percorrido por diferentes solucións baseadas en DL para a clasificación de imaxes multi e hiperespectrais que apoie a seguinte hipótese:

> *É posible deseñar esquemas de clasificación simples para imaxes multi e hiperespectrais baseados en DL, que utilicen información espacial e de texturas para lograr bos resultados en termos de precisión, e que se poidan executar en tempo real utilizando infraestruturas computacionais de baixo custo.*

O éxito de esquemas eficientes requirirá dun estudo a fondo para atopar técnicas que melloren a precisión da clasificación adaptadas ao modelo de computación deste hardware de baixo custo, no noso caso CPUs (*Central Processing Units*) multicore e GPUs.

As contribucións principais desta tese son as seguintes:

1. **Proposta dun esquema de clasificación espectral-espacial baseado en redes neuronais con pesos aleatorios.** KELM-EMP é un esquema baseado en extracción de características, fusión de datos e KELM. O esquema consiste nunha primeira fase de extracción de características onde, tras unha redución da dimensionalidade da imaxe empregando un algoritmo PCA, extráese a información espacial presente na imaxe mediante o uso de MPs. A información espectral e a información espacial fusiónanse mediante unha suma ponderada, permitindo axustar a influencia de cada tipo de información no clasificador. A clasificación final lévase a cabo mediante un clasificador baseado en KELM. A novidade deste esquema é o uso dunha rede neuronal como clasificador, cuxa fase de adestramento estándar foi substituída pola asignación de pesos mediante o uso dunha función kernel. Ademais, engadiuse unha etapa final de regular-

ización espacial que permite reducir o número de píxeles clasificados de forma illada e aumentar a precisión.

2. **Proposta dun esquema baseado en texturas, para a identificación de especies vexetais sobre imaxes multiespectrais capturadas por vehículos aéreos non tripulados (*UAVs, Unmanned Aerial Vehicles*).** Trátase dun esquema baseado en extracción de características e un sistema de clasificación xerárquico formado por un conxunto de máquinas de soporte vectorial (*SVMs, Support Vector Machines*). Neste esquema utilízanse, ademais da propia información espectral, información espacial e de texturas. A extracción de información espacial realízase mediante operacións morfolóxicas, mentres que para a información de texturas utilízase *k*-means [29, 30] e VLAD [31].

   As imaxes obtidas por UAVs adoitan posuír unha gran densidade de información, o que incrementa o tempo de procesamento. Tratando de reducir ese tempo, neste esquema utilízase o algoritmo de superpíxeles SLIC. Este algoritmo agrupa os píxeles que posúen trazos similares dentro dunha imaxe, xerando un mapa de segmentación. Desta forma, para a extracción de características, en lugar de traballar directamente con cada un dos píxeles da imaxe, utilízanse os centroides (ou representantes) de cada un dos superpíxeles.

   Un clasificador xerárquico composto por catro SVMs encárgase da clasificación final. Os dous primeiros SVMs separan as clases en grandes grupos mentres que os dous últimos realizan unha clasificación máis específica. A novidade deste esquema é, por unha banda, a utilización de superpíxeles en combinación coa información espectral-espacial e de texturas e, pola outra, o uso dun clasificador xerárquico.

3. **Proposta de esquemas de clasificación baseado en CNNs.** Neste caso o traballo consistiu no estudo preliminar de CNNs mediante a comparativa de diferentes posibilidades de implementación. Utilizáronse diferentes librerías para a execución dun esquema de clasificación baseado en CNNs mediante o framework de DL chamado Caffe [39]. Despois de levar a cabo unha redución de dimensionalidade sobre a imaxe a tratar, extráese da imaxe os patches correspondentes ás mostras de adestramento que serán utilizadas pola CNN para aprender os coeficientes dos filtros convolucionais. A achega desta análise era a de coñecer ata que punto o uso de librerías optimizadas permitían reducir o tempo de execución nun esquema baseado en CNNs.

4. **Proposta dun esquema de clasificación baseado en CNNs e TCA para DA.** O esquema denominado TCANet que se propón baséase no uso dunha CNN para extraer características comúns de imaxes pertencentes a dominios diferentes. Dita CNN non utiliza o procedemento estándar de adestramento, baseado na propagación de pesos, senón que este foi substituído polo algoritmo TCA para calcular os pesos dos dous filtros convolucionais que compoñen a rede do esquema. Ao comezo do esquema realízase un aliñamento condicional da covarianza das dúas imaxes, pero só se isto consegue reducir a distancia entre as distribucións das dúas imaxes.

   A novidade deste esquema é, por tanto, o uso da técnica TCA para calcular os coeficientes da rede convolucional. Redúcese así o custo de aprendizaxe que leva o uso doutras técnicas baseadas na propagación de pesos e o uso de gradientes.

5. **Implementación eficiente de esquemas de clasificación en CPUs multinúcleo e GPUs usando OpenMP e CUDA (*Compute Unified Device Architecture*) respectivamente:**

   - Dous esquemas, un baseado en ELM (ELM-EMP) e outro baseado en KELM (KELM-EMP) foron desenvolvidos en tres diferentes versións: CPU, OpenMP e GPU. Estudáronse distintas estratexias para particionar os datos e diferentes configuracións de bloques co fin de explotar ao máximo a capacidade computacional da GPU. Tamén se fixo uso de librerías optimizadas.
   - No esquema baseado en texturas para clasificación de imaxes multiespectrais de vexetación utilizáronse estratexias de paralelización mediante OpenMP e GPU que permitiron reducir o tempo de execución.
   - O esquema TCANet desenvolvido para DA executouse sobre Matlab, facendo uso do conxunto de ferramentas de computación paralela dispoñibles neste software. Isto permite paralelizar a execución do esquema sobre cada un dos píxeles da imaxe.
   - Os esquemas Caffe-HYCNN e cuDNN-HYCNN son dúas implementacións en GPU do esquema HYCNN. Ambos esquemas son executados utilizando o framework de DL Caffe e funcións optimizadas da librería cuDNN (*CUDA Deep Neural Network*).

6. **Desenvolvemento dunha aplicación de escritorio para a análise e procesamento de imaxes de sensado remoto.** Este software, denominado HypeRvieW, é unha ferramenta de escritorio que permite analizar as características de imaxes de sensado remoto

e aplicar algoritmos de procesado sobre as mesmas, incluíndo rexistrado ou clasificación entre outras tarefas. En referencia á análise, o software é capaz de mostrar as diferentes bandas que forman a imaxe así como tamén de visualizar a firma espectral de cada un dos seus píxeles. Do mesmo xeito, permite comparar as firmas espectrais de dous píxeles entre si ou cunha colección de firmas espectrais pertencentes a diferentes materiais. O software dispón das ferramentas básicas de visualización de imaxes como son zoom, desprazamento a unha determinada banda, rotación, escalado e recorte da imaxe.

En canto ao procesamento de imaxes, HypeRvieW permite xerar esquemas de clasificación supervisada combinando os diferentes módulos presentes no propio software ou mediante a incorporación de novos módulos desenvolvidos polo usuario. Entre os módulos dispoñibles podemos destacar un algoritmo de redución da dimensionalidade baseado en PCA; un algoritmo de extracción de características espaciais baseado en operacións morfolóxicas; algoritmos de segmentación como SLIC, RQS (*Really Quick Shift*) [40] e Watershed [41, 42]; algoritmos de rexistrado baseados na transformada rápida de Fourier (*FFT, Fast Fourier Transform*) e a transformada de Fourier fraccional multicapa (*MFFT, Multilayer Fractional Fourier Transform*) [43]; ou algoritmos de clasificación baseados en ELM e SVM. Unha importante característica desta ferramenta é que permite ao usuario incorporar os seus propios desenvolvementos ao software e utilizalos para xerar novos esquemas de clasificación.

Por último, é importante destacar que HypeRvieW facilitounos mostrar a empresas, estudantes e público en xeral, dunha maneira simple, as posibles aplicacións da investigación desenvolvida polo grupo de investigación especializado en procesado de imaxe multi e hiperespectral, liderado polos directores desta tese de doutoramento.

USC
UNIVERSIDADE
DE SANTIAGO
DE COMPOSTELA

Summary

The objective of this thesis is the development of efficient schemes, based on the use of deep learning neural networks, for the classification of remotely sensed multi and hyperspectral land cover images. Efficient schemes are those that are capable of obtaining good results in terms of classification accuracy and that can be computed in a reasonable amount of time depending on the task performed. Regarding computational platforms, multicore architectures and Graphics Processing Units (GPUs) will be considered.

Remote sensing can be defined as the acquisition of information from a given scene without coming into physical contact with it, through the use of sensors, mainly located on aerial platforms, which capture information in different ranges of the electromagnetic spectrum. Depending on the sensor used for data acquisition, the reflectance values captured can range from near infrared to ultraviolet, also including the visible spectrum. The dataset acquired is defined by the size of the spectral range captured, so if the range includes only a few discrete frequencies, the data set is called a multispectral image, whereas if the range is wider and usually continuous, on the order of a hundred bands or more, the data set will be called a hyperspectral image. This also influences the techniques to be selected for data processing, since a larger number of bands implies a larger amount of information to be processed and, therefore, more potential for extracting relevant information as well as greater computational complexity.

Each of the different materials we can observe has a spectral signature, i.e. a set of unique reflectance values that characterizes it. The special characteristics of hyperspectral images, which provide detailed information on each of the pixels that make up the image, allow the distinction between physical materials and objects even at the pixel level. This distinction will be more accurate the larger the number of bands in the captured image. The high dimensionality of the data in this type of images presents new challenges in techniques such as unmixing

[1, 2], target and anomaly detection [3, 4], feature extraction [5, 6] or characterization and classification of elements on the Earth's surface [5, 7].

The classification of images of the Earth surface is one of the most common processes. During this process, each of the pixels that make up the image is classified, that is, it is assigned a class label, according to a set of previously defined classes. This classification can be supervised or unsupervised. During unsupervised classification the algorithm works without any information other than what it is able to extract from the data set itself. On the other hand, supervised classification needs some reference information, i.e. it needs to know to which class some of the image pixels belong. This information will be used during a learning process that will allow the algorithm to acquire the necessary knowledge to classify the rest of the image.

Machine Learning (ML)-based classification algorithms play a very important role in the remote sensing literature [8]. ML encompasses a set of algorithms that capture system dynamics and do not require human intervention to make certain changes. This makes them more robust and less dependent on human experts. Meanwhile, Deep Learning (DL) defines a subset of ML algorithms, which learn to represent the problem as a nested hierarchy of concepts where more general concepts are defined in relation to simpler concepts and more abstract representations are obtained based on less abstract ones.

ML-based architectures such as Artificial Neural Networks (ANNs) [9, 10] or architectures based on DL such as Deep artificial Neural Networks (DNNs) or Convolutional Neural Networks (CNNs) [11, 12, 13] have been successfully used for remote sensing image classification. The standard learning method for this type of networks consists of an iterative process, which is repeated until a certain number of iterations is reached or until the error produced by the network, i.e. the difference between the output obtained by the network and the desired output, is reduced below a certain threshold. Depending on the network configuration, this learning process can be very costly.

Schmidt et al. [14] proposed a method based on random weights as an alternative to standard learning up to that time, which reduced the computational cost. This idea was later applied by Huang et al. [15] to develop an algorithm called Extreme Learning Machine (ELM). The basic idea was to randomly initialize the weights of the hidden layers of a neural network so that only the weights of the output layer needed to be calculated. Based on the least-squares solution of a general linear system and the Moore-Penrose inverse calculation, ELM emerged as an alternative to the standard learning approach based on weight propagation. Later, Huang

et al. [16] proposed the use of kernel-based functions to replace the random assignment of weights. They thus tried to avoid the differences between the results obtained by repeating the same experiments, due to the randomness of the weights. This new version of ELM was called Kernel-based Extreme Learning Machine (KELM).

Hyperspectral images contain an enormous amount of spectral information that can become redundant. So-called feature extraction methods are often used to extract the most relevant information. Their application results in image dimensionality reduction and among them are methods such as Autoencoders (AE) [17], Independent Component Analysis (ICA) [18], or Principal Component Analysis (PCA) [19]. PCA is one of the most widely used methods in the literature for this purpose [20] and allows the dimensionality of a dataset to be reduced while preserving as much variation present in the data as the user indicates.

In addition to the spectral information, which is implicit in the image, there is another type of information, the spatial information, which can be extracted from the data sets. This type of information usually refers to the environment surrounding each of the pixels that make up the image. Different methods such as segmentation [21], which performs an exhaustive partitioning of the image into regions based on a homogeneity criterion, or methods based on fixed structures such as Morphological Profiles (MPs) [5] can be used to extract spatial information. The MPs extract spatial information from the nearest neighbors of a given pixel by morphological operations called opening and closing, while preserving the geometric features of the structures in the image. These operations use structuring elements of different sizes that traverse the image to form a set of MPs called the Extended Morphological Profile (EMP) [22, 23]. This EMP represents the variability at different scale levels of the structures contained in the image.

There are certain cases, such as the detection of defects in surfaces [24], the analysis of medical images [25] or the analysis of vegetation [26] among others, in which the use of spatial and spectral information may not be sufficient to discriminate between elements. In these cases, texture information (which is still spatial) can help to improve class separability and thus increase classification accuracy. The texture of an image is related to the spatial distribution of intensity values in the image and, as such, contains information about contrast, uniformity, roughness, etc., which allows characterizing the different regions of interest within a scene. There is a wide range of algorithms used in the extraction of texture information: Gaussian Mixture Modeling (GMM) [28], or k-means [29, 30] are some examples. For both

GMM and *k*-means, the goal is to generate a dictionary of primitives called textons, which constitute the texture.

With the generated texton dictionary, we can make use of feature encoding algorithms such as Vector of Locally Aggregated Descriptors (VLAD) [31] to characterize previously defined regions of an image as textures based on the individual characteristics of each of the pixels in those regions. One of the most common techniques used for extracting uniform regions in images is segmentation based on superpixel [32, 33]. Superpixels provide a compact representation of images, allowing processing algorithms to be applied to segments rather than individual pixels, thus reducing the computational cost of the algorithms while maintaining the quality of the solution obtained. In particular, the Simple Linear Iterative Clustering (SLIC) [34] algorithm allows defining superpixels of similar size and regularity throughout the entire image.

The resolution of classification problems using supervised methods, such as those proposed in this thesis, is strongly linked to the existence of reference information. In many cases, building this information is costly in time and effort since it may require data from areas of interest of difficult access and contrasting information from different sources. In the case of the vegetation images analyzed in this doctoral thesis, it has been necessary to travel together with experts from "Aguas de Galicia" to the edge of the rivers, to contrast the information provided by them with regional vegetation inventories and images obtained by tools such as Google Earth and finally to perform an iterative process of refining the reference information using software tools developed by us. In addition, if we add to this the large amount of new images acquired every day for use with land cover applications, we are faced with a task that is sometimes intractable.

On the other hand, if the classification problem to be solved involves images corresponding to different geographical areas, or captured by different sensors or in different time periods, the task becomes more complex because the learning obtained for one image cannot be directly extrapolated to others. There is also spectral displacement between spectral images due, for example, to atmospheric effects, temperature or even the sensor itself in charge of capturing the data, which must be taken into account. Faced with this problem, the need arises to design classification schemes that are able to take advantage of the scarce reference information available and extract knowledge from the classification of some images to be used for the classification of images obtained, for example, in other geographical areas.

Transfer Learning (TL) encompasses a set of techniques that are responsible for applying the knowledge previously acquired for one or more tasks in a given source domain, to another related to the initial task, in a target domain [35]. In turn, Domain Adaptation (DA) techniques, which are a subset of TL, use reference information from images belonging to a source domain to try to classify images belonging to a target domain, for which no reference information is available. Among the techniques applied to achieve DA we can highlight the Feature Extraction (FE) techniques [36, 37], which try to reduce the displacement between the two domains by finding a function that can map the data to a new space that better defines them.

The use of DL-based architectures, such as CNNs, also allows finding the necessary mapping function to be able to apply the knowledge of one domain to the other, i.e., to perform DA. This type of networks is basically formed by a set of nested convolutional filters that extract features in a hierarchical way. As mentioned above, these neural networks require an iterative process to learn the weights of the network, which, depending on the complexity of the network, can be very expensive. Therefore, different alternatives have been proposed, on this type of neural networks, to reduce this cost, such as [38], to which we have paid special attention in this thesis.

In particular, it has been proposed to replace the costly iterative process of learning a CNN by computing the convolutional network filters using the technique called Transfer Component Analysis (TCA). TCA is a feature extraction technique specially designed for DA. The goal of TCA is to condition the data distributions between domains in a new space, allowing to preserve the properties of the data and at the same time reduce the deviation between the distributions. By applying TCA to the computation of the filters of the CNN network, a reduction of the computational cost is achieved while reducing the displacement between domains, i.e., the CNN network is adapted to the DA problem.

Given the complexity of the classification techniques and the large amount of data available in the images with which we have worked, and in view of all the above, we propose in this thesis a tour of different solutions based on DL for the classification of multi and hyperspectral images to support the following hypothesis:

> *It is possible to design simple classification schemes for multi and hyperspectral images based on DL, which use spatial and texture information to achieve good results in terms of accuracy, and which can be run in real time using low-*

cost computational infrastructures.

The success of efficient schemes will require an in-depth study to find techniques to improve classification accuracy adapted to the computational model of this low-cost hardware, in our case multicore Central Processing Unit (CPUs) and GPUs.

The main contributions of this thesis are the following:

1. **Proposal of a spectral-spatial classification scheme based on neural networks with random weights**. KELM-EMP is a scheme based on feature extraction, data fusion and KELM. The scheme consists of a first phase of feature extraction where, after a dimensionality reduction of the image using a PCA algorithm, the spatial information present in the image is extracted using MPs. The spectral information and the spatial information are merged by means of a weighted sum, allowing to adjust the influence of each type of information in the classifier. The final classification is carried out using a classifier based on KELM. The novelty of this scheme is the use of a neural network as a classifier, whose standard training phase has been replaced by the assignment of weights using a kernel function. In addition, a final spatial regularization stage has been added to reduce the number of pixels classified in isolation and increase accuracy.

2. **A proposed texture-based scheme for the identification of plant species on multispectral images captured by Unmanned Aerial Vehicles (UAVs).** This is a scheme based on feature extraction and a hierarchical classification system formed by a set of Support Vector Machines (SVMs). In this scheme, in addition to the spectral information itself, spatial and texture information are used. The extraction of spatial information is performed by morphological operations, while texture information is extracted using k-means [29, 30] and VLAD [31].

   The images obtained by UAVs usually have a high density of information, which increases the processing time. In an attempt to reduce this time, the SLIC superpixel algorithm is used in this scheme. This algorithm groups pixels with similar features within an image, generating a segmentation map. Thus, for feature extraction, instead of working directly with each of the image pixels, the centroids (or representatives) of each of the superpixels are used.

   A hierarchical classifier consisting of four SVMs is in charge of the final classification. The first two SVMs separate the classes into large groups while the last two perform

a more specific classification. The novelty of this scheme is on the one hand the use of superpixels in combination with spectral-spatial and texture information and on the other hand the use of a hierarchical classifier.

3. **Proposal of classification schemes based on CNNs.** In this case the work consisted in the preliminary study of CNNs by comparing different implementation possibilities. Different libraries were used for the execution of a classification scheme based on CNNs using the DL framework called Caffe [39]. After performing a dimensionality reduction on the image to be treated, the patches corresponding to the training samples that will be used by the CNN to learn the coefficients of the convolutional filters are extracted from the image. The contribution of this analysis was to determine to what extent the use of optimized libraries allowed reducing the execution time in a scheme based on CNNs.

4. **Proposal of a classification scheme based on CNNs and TCA for DA.** The proposed so-called TCANet scheme is based on the use of a CNN to extract common features from images belonging to different domains. This CNN does not use the standard training procedure, based on the propagation of weights, it has been replaced by the TCA algorithm to calculate the weights of the two convolutional filters that compose the network of the scheme. At the beginning of the scheme a conditional alignment of the covariance of the two images is performed, but only if this succeeds in reducing the distance between the distributions of the two images.

   The novelty of this scheme is, therefore, the use of the TCA technique to calculate the coefficients of the convolutional network. This reduces the learning cost of using other techniques based on the propagation of weights and the use of gradients.

5. **Efficient implementation of classification schemes on multicore CPUs and GPUs using OpenMP and Computer Unified Device Architecture (CUDA) respectively:**

   - Two schemes, one based on ELM (ELM-EMP) and one based on KELM (KELM-EMP) have been developed in three different versions: CPU, OpenMP and GPU. Different data partitioning strategies and different block configurations have been studied in order to exploit the computational capacity of the GPU to the maximum. Optimized libraries have also been used.

- In the texture-based scheme for classification of multispectral vegetation images, parallelization strategies using OpenMP and GPU have been used to reduce the execution time.
- The TCANet scheme developed for DA has been executed on Matlab, making use of the set of parallel computing tools available in this software. This allows parallelizing the execution of the scheme on each pixel of the image.
- The Caffe-HYCNNN and cuDNNN-HYCNNN schemes are two implementations in GPU of the HYCNNN scheme. Both schemes are executed using the Caffe DL framework and optimized functions from the CUDA Deep Neural Network (cuDNN) library.

6. **Development of a desktop application for the analysis and processing of remote sensing images.** This software, called HypeRvieW, is a desktop tool that allows to analyze the characteristics of remote sensing images and apply processing algorithms on them including registration or classification among other tasks. In reference to the analysis, the software is able to display the different bands that form the image as well as to visualize the spectral signature of each of its pixels. Likewise, it allows comparing the spectral signatures of two pixels with each other or with a collection of spectral signatures belonging to different materials. The software has the basic image viewing tools such as zoom, panning, rotation, scaling and cropping of the image.

   Regarding image processing, HypeRvieW allows the generation of supervised classification schemes by combining the different modules present in the software itself or by incorporating new modules developed by the user. Among the available modules we can highlight a dimensionality reduction algorithm based on PCA; a spatial feature extraction algorithm based on morphological operations; segmentation algorithms such as SLIC, Really Quick Shift (RQS) [40] and Watershed [41, 42]; registration algorithms based on the Fast Fourier Transform (FFT) and Multilayer Fractional Fourier Transform (MFFT) [43]; or classification algorithms based on ELM and SVM. An important feature of this tool is that it allows users to incorporate their own developments into the software and use them to generate new classification schemes.

   Finally, it is important to highlight that HypeRvieW has allowed us to present to companies, students, and the general public, in a simple way, the possible applications of

the research carried out by the research group specialized in multi and hyperspectral image processing, led by the directors of this doctoral thesis.

Contents

USC
UNIVERSIDADE
DE SANTIAGO
DE COMPOSTELA

Acronyms

AA Average Accuracy.

AD Allocation Disagreement.

AE Autoencoder.

AF Attribute Filter.

ANN Artificial Neural Network.

AP Attribute Profile.

API Application Programming Interface.

AVIRIS Airborne Visible/Infrared Imaging Spectrometer.

BoW Bag of Words.

CC Compute Capability.

CHM Canopy Height Model.

CNN Convolutional Neural Network.

CPU Central Processing Unit.

CS Class-Specific.

CUDA Compute Unified Device Architecture.

cuDNN CUDA Deep Neural Network.

DA Domain Adaptation.

DAE Denoising Autoencoder.

DANN Domain-Adversarial Neural Network.

DBN Deep Belief Network.

DL Deep Learning.

DNN Deep Neural Network.

DRAM Dynamic Random Access Memory.

EF Extinction Filter.

ELM Extreme Learning Machine.

EMP Extended Morphological Profile.

EP Extinction Profile.

EVD Eigenvalue Decomposition.

FE Feature Extraction.

FOV Field of View.

FV Fisher Vector.

GAN Generative Adversarial Network.

GCC GNU Compiler Collection.

GIS Geographic Information System.

GMM Gaussian Mixture Model.

GPGPU General-Purpose Graphic Processing Unit.

GPU Graphics Processing Unit.

GRASS GIS Geographic Resources Analysis Support System Geographic Information System.

IFOV Instantaneous Field of View.

KELM Kernel-based Extreme Learning Machine.

ML Machine Learning.

MP Morphological Profile.

NDVI Normalized Difference Vegetation Index.

NNRW Neural Networks with Random Weights.

NVCC NVIDIA CUDA Compiler.

OA Overall Accuracy.

OMP Open Multi-Processing.

PC Personal Computer.

PC Principal Component.

PCA Principal Component Analysis.

QD Quantity Disagreement.

RBM Restricted Boltzmann Machine.

ROSIS-03 Reflective Optics System Imaging Spectrometer.

SAE Stacked Autoencoder.

SDAE Stacked Denoising Autoencoders.

SE Structuring Element.

SIMD Single Instruction, Multiple Data.

SIMT Single Instruction, Multiple Thread.

SLFN Single hidden Layer Feedforward neural Network.

SLIC Simple Linear Iterative Clustering.

SM Streaming Multiprocessor.

SVM Support Vector Machine.

TCA Transfer Component Analysis.

TIFF Tag Image File Format.

TL Transfer Learning.

UAV Unmanned Aerial Vehicle.

VAE Variational Autoencoder.

VLAD Vector of Locally Aggregated Descriptors.

CHAPTER 1

INTRODUCTION

1.1 Motivation

In this section the theoretical framework for the different topics addressed in this thesis is presented. The thesis focuses on the development of efficient techniques for processing multispectral and hyperspectral images using Deep Learning (DL) techniques over low-cost computing infrastructures. The proposed techniques are compared in terms of accuracy for solving a classification task, and in terms of execution time.

We will start this section by introducing multilayer perceptron neural networks, as well as Extreme Learning Machine (ELM) and Kernel-based Extreme Learning Machine (KELM) networks, derived from the previous ones. They are used in remote sensing to process spectral information, although in many cases some preprocessing stages are included that aim to extract spatial information to be considered together with the spectral one. Next, we will introduce classification methods based on textures, that are very efficient for the classification of vegetation such as detecting the structure patterns that characterize the different species. We will continue with Convolutional Neural Network (CNN), the most extended model of DL networks. Then, a network model (PCANet) that has been proposed to simplify the training process by performing domain adaptation will be presented. Finally, some implementation features of DL algorithms and the benefits they provide to the field of remote sensing will be explained.

Artificial Neural Networks (ANNs) are computational models inspired by how the biological neural networks of the human brain process information [44]. They became very popular thanks to state-of-the-art results achieved on many visual recognition tasks. The neuron is the

basic unit of computation in a neural network. It generates an output based on inputs (one or more) coming from other neurons or from external sources. Each input has an associated weight that is assigned based on the importance over the other inputs. Furthermore, an activation function is used to introduce non-linearity into the output of a neuron, thus being able to learn the non-linearity present in most of the real world data.

The simplest ANN is called perceptron [45]. This single-layer neural network contains an input layer in charge of supplying the input data to this neuron and a computational output layer with a single neuron. Due to the fact that the output generated by a perceptron is based on a threshold function, it is considered a binary classifier. On the other hand, a multilayer neural network contains more than one computational layer. Three types of layers can be identified: an input layer, that receives the external information, a hidden layer, that processes the information coming from a previous layer and communicate the result to the next one, and an output layer [46].

ANNs manage two kinds of information. The first one is volatile and it is the input data used to train the neural network. The second one is non volatile and it is used to store the patterns learned by the neural network through the weights and biases. Finally, in terms of learning, ANNs can be classified into three types [47, 48]: supervised learning, where the input data and the target output are used to adjust the weights and biases of the neural network; unsupervised learning, where only the input data is available and the neural network should classify it as a function of their common features; and reinforcement learning, that is a hybrid learning between the supervised and unsupervised learning, rewarding the network when it hits the mark and penalizing it when it misses.

With respect to a multilayer feedforward neural network with supervised learning, its training can be defined as an iterative process with two main steps: feedforward and backward. In the feedforward step, the input layer is fed with each of the samples available in the input data. These samples are processed by the hidden layers of the neural network and finally, the output layer produces the final value. At this point the error between the target value and the value obtained by the neural network is computed using a loss function. Then, the backward step, where learning is actually carried out, begins. The main idea is to minimize the error produced at the output of the network by adjusting the weights and biases of the neural network that where randomly initialised in the first iteration. For this step, the most used technique is the gradient descent algorithm [49, 50]. The iterative process is repeated until the error is less than a certain threshold or the number of iterations reach a certain value.

In addition to the gradient descent method, other training schemes using random weights have been proposed [51]. One of the most attractive theories was presented in 2004 with a new learning algorithm, called ELM [52], for Single hidden Layer Feedforward neural Networks (SLFNs). A scheme is proposed in which the weights of the hidden units are fixed and randomly assigned is proposed. So, the learning problem is reduced to calculating the weights of the output layer, finding a least-squares solution for a simple linear system.

Despite having been the subject of discussions about its originality, the truth is that ELM lays the foundations for the development of other methods such as KELM, that obtains very good classification results at a lower computational cost. KELM uses a kernel function to obtain the feature mapping (the weights of the hidden layer). The output weights are computed by using the Moore-Penrose generalized inverse to solve a simple liner system.

Hyperspectral images contain both spectral and spatial information that can be exploited during the processing. When an algorithm takes into account both kinds of information, the results show accuracy improvements compared to algorithms that only use one of them [22, 53, 54]. Mathematical morphology [55] is one of the most common techniques used to extract spatial information from hyperspectral images such as, for example, the size, orientation, and contrast of the spatial structures present in the image. Different methods for extracting spatial information were, for example, Attribute Profiles (APs) [56] and Extinction Profiles (EPs) [57]. In the case of APs, the introduction of multiple different attributes via Attribute Filters (AFs) for the characterization of the image allows for a more accurate modeling of the scene. EPs [57] make use of Extinction Filters (EFs) to create a fully automatic method using regional extrema of attributes. EPs were shown to provide better classification results than APs. In this thesis, a computationally efficient KELM-based classification scheme that combines spectral and spatial information extracted by an Extended Morphological Profile (EMP) is proposed.

In some cases, such as the classification of vegetation species, in addition to the spectral-spatial information, the textural information also helps to discriminate between different species with very similar spectral signatures [27]. Texture refers to the spatial organization of a set primitives called textons [58]. They are related to fundamental micro-structures in natural images and are considered as the atoms of pre-attentive human visual perception [59]. Texture representations based on Bag of Words (BoW) [60] and on CNNs [61] have been extensively studied with impressive performance. Usually, k-means [60] or Gaussian Mixture

Models (GMMs) [28] are used to generate a visual vocabulary based on which histograms of visual words are computed in order to quantize and represent the input data.

Vegetation analysis plays an important role in the context of current climatic change conditions [62], e.g. to control the water resources or to monitor vegetation changes [63, 64]. Information about the number of different vegetation species present in a region, and their interaction with the environment can be extracted from the analysis of remote sensing data [65]. In some cases, the use of satellite-based solutions is a cost effective solution. But, they present some problems like revisiting time or data capture on cloudy days. These problems are avoided using Unmanned Aerial Vehicles (UAVs) as they provide a much more flexible and affordable platform with a higher spatial resolution at lower cost [26]. Although they are not adequate for all the applications, the wide availability of images taken by sensors on UAVs allow even small companies use them. In this thesis, a texture-based classification scheme for the identification of vegetation species and non-vegetation structures from multispectral images captured by a sensor onboard a UAV is proposed.

In recent years, DL [66], an area within Machine Learning (ML) [67, 8], has caught the attention of the community and has been widely used in the field of classification of hyperspectral datasets [68, 69, 70, 66]. Compared to traditional methods, DL techniques extract features from the original data using a large set of hierarchical layers that form a neural network [71]. These layers can be trained both in a supervised or in a unsupervised manner. The network layers comprise non-linear processing units that extract and transform the data coming from a previous layer, and send it to the next layer of the network. That structure allows DL neural networks to learn features or represent data at different levels of complexity. The higher-level features are derived from those of lower-levels to form a hierarchical representation. The deeper the network is, the more complex the extracted characteristics are. At the same time, each level of this hierarchical representation corresponds to a different level of abstraction, building a hierarchy of concepts. DL techniques are applied to problems such as classification [72] and Domain Adaptation (DA) [73, 74] among others, topics that will be explored in this thesis.

Different DL-based architectures are, for example, Deep Neural Networks (DNNs) [75], CNNs [76], and Deep Belief Networks (DBNs) [77], among many others. A description of some of them is detailed in what follows, focusing on those that have been used in this thesis.

One of the main assumptions in many ML algorithms is that the training and test data must be in the same feature space and have the same distribution to perform a successful classifi-

cation [78]. But, on real world problems, these conditions are rarely met. So, new methods following a different perspective need to be proposed. Transfer Learning (TL) techniques are focused on applying the knowledge previously acquired for one or more source tasks for a source domains to a similar or different but related target task over a target domain [35]. These techniques could be grouped into three categories [78]:

1. Unsupervised. In this group no matter whether the domains are the same or not, the target task is different from but related to the source task, and no labelled information is available in both source and target domains for training.

2. Inductive. In this category the domains are not important, they can be the same or not. Nevertheless, both the source and the target tasks are different. Moreover, some labelled samples from the target domain are compulsory to build an objective predictive model.

3. Transductive. In this case both, the source and the target domains, are different, while the source and the target tasks are the same. For this category no labelled data coming from the target domain are required. On the contrary, a large number of labelled samples from the source are required.

DA is a particular case of TL. It uses the labelled data from images belonging to one or more related source domains to classify unlabelled images from target domains. To tackle DA problems, most of the recent literature focuses on representation learning or Feature Extraction (FE) [79, 73]. The main objective of this technique is to find a mapping function that converts the input data belonging to both domains (source and target) from its original representation space to a new space that better describes it [79]. This is an important stage to build a successful classifier model, since its performance depends strongly on the selection of a good representation of the input data [80]. As examples of DA techniques related with FE we can mention modern neural networks approaches such as CNNs [81], Denoising Autoencoders (DAEs) [82, 83] or Domain-Adversarial Neural Networks (DANNs) [84, 85, 86].

CNNs are biological inspired by the visual cortex [87]. Their main objective is to learn higher-level features in the data through convolutions. Similar to ANNs, CNNs require a procedure where the weights of the network have to been learned based on a iterative training process. Depending on the number of samples available and the complexity of the network, the training process can be very time consuming. Thus, less expensive computational techniques for training CNNs have been proposed. A scheme that uses fixed weights is presented

in [38] where the author proposes PCANet. It is a linear network for classification composed of two stages where the weights are fixed and computed using Principal Component Analysis (PCA) filters [19], which considerably reduces the computational cost. PCANet has been aplied to DA [88].

In [89], the authors use a method called Transfer Component Analysis (TCA) to learn some transfer components across different but related domains (source and target domains). So, the new representation of the features can be applied to traditional machine learning methods to train a classification model using the source data and then use it on the target data. TCAs is a kernel-based feature extraction technique especially designed for DAs. TCAs aims to match data distributions between domains in a latent space instead of using the original feature space, so the data properties can be preserved. In this thesis, a classification scheme for DA, named TCANet, based on a CNN with fixed weights computed by TCA is proposed.

The computational complexity of the algorithms for processing multi and hyperspectral images in general and DL-based classification algorithms in particular, such as CNNs, makes it necessary to use architectures that allow their parallelization. Graphics Processing Units (GPUs)-based architectures and multicores are examples of low-cost platforms for this purpose. In the case of GPUs the increasing development of frameworks that allow combining the strong capacity of deep network models with their the computational power helps to build computationally efficient classification schemes in an easier way. TensorFlow [90], PyTorch [91], Caffe [92], and Keras [93] are some examples of currently active frameworks and some of the most used.

In this thesis, all the techniques were designed trying to reduce the computation time. For example, non iterative training methods are considered for the KELM and TCANet proposals. Different proposals for efficient spectral-spatial DL-based classification on low-cost computing infrastructures, such as multi-threaded Central Processing Units (CPUs) and many-core GPUs shipped in personal computers, are also presented. The most relevant contributions in this line are a CNN-based classification scheme using the Caffe framework and a KELM-based one using Compute Unified Device Architectures (CUDAs) [94].

1.2 Hypothesis and objectives

Provided the motivation presented in the previous section, the hypothesis analyzed in this thesis is the following:

It is possible to design simple classification schemes for multi and hyperspectral images based on DL, which use spatial and texture information to achieve good results in terms of accuracy, and which can be run in real time using low-cost computational infrastructures.

Based on the hypothesis, the following main objective is formulated:

- Explore and develop computationally efficient classification schemes for multispectral and hyperspectral images, based on different DL architectures, such as ANNs and CNNs, that improve classification accuracy over existing schemes. As the number of DL alternatives is high, a detailed analysis of the literature will be required. In addition, the large amounts of data available in the hyperspectral images makes a careful design for efficient computation in low-cost computing platforms necessary.

The detailed objectives are the following:

- *Develop a spectral-spatial classification scheme based on SLFNs with random weights as a simple approximation, as well as an efficient implementation to be executed on GPUs.* The scheme will comprise a first step to reduce the dimensionality of the dataset, followed by morphological operations to extract spatial information. Then, the extracted Morphological Profiles (MPs) combined with the original image will be processed by a KELM-based classifier to produce the final classification map.

  This objective is addressed in the following papers:

  - A. S. Garea, D. B. Heras, and F. Argüello, "GPU classification of remote-sensing images using kernel ELM and extended morphological profiles," *International Journal of Remote Sensing*, vol. 37, no. 24, pp. 5918–5935, 2016
  - A. S. Garea, D. B. Heras, and F. Argüello, "Clasificación de imágenes de teledetección mediante ELM kernel y perfiles morfológicos en GPU," in *Jornadas SARTECO 2016*. Ediciones Universidad de Salamanca, 2016, pp. 81–89

- *Develop a texture-based classification scheme for the identification of vegetation species and non-vegetation structures in multispectral datasets.* The scheme will use a hierarchical classifier that leverages the spectral, spatial, and texture information extracted from the dataset to identify the different elements.

  This objective is addressed in the following paper:

  - P. G. Bascoy, A. S. Garea, D. B. Heras, F. Argüello, and A. Ordóñez, "Texture-based analysis of hydrographical basins with multispectral imagery," in *Remote Sensing for Agriculture, Ecosystems, and Hydrology XXI*, C. M. U. Neale and A. Maltese, Eds., vol. 11149, International Society for Optics and Photonics. SPIE, 2019, pp. 225 – 234

- *Propose a CNN-based classification scheme to be executed on GPU*. A comparison between a CPU and a GPU version will be performed.

  This objective is addressed in the following papers:

  - A. S. Garea, D. B. Heras, and F. Argüello, "Caffe CNN-based classification of hyperspectral images on GPU," *The Journal of Supercomputing*, vol. 75, no. 3, pp. 1065–1077, 2019
  - A. S. Garea, D. B. Heras, and F. Argüello, "GPU classification for hyperspectral images based on convolutional neural networks," in *Computational and Mathematical Methods in Science and Engineering*, J. Vigo-Aguiar, Ed., 2017, pp. 912–923

- *Develop a CNN-based classification scheme, without backpropagation, for DA.* This scheme will include a FE step carried out by a CNN with fixed weights computed based on TCA, and a final classification step.

  This objective is addressed in the following paper:

  - A. S. Garea, D. B. Heras, and F. Argüello, "TCANet for domain adaptation of hyperspectral images," *Remote Sensing*, vol. 11, no. 19, p. 2289, 2019

- *Propose a free desktop modular application for hyperspectral remote-sensing image processing that allows the user to inspect the data in detail, to build reference data for classification, to register pairs of images, and to perform the segmentation or classification of an image.* The tool will also allow the user to introduce her own algorithms for carrying out the different tasks.

  This objective is addressed in the following papers:

  - A. S. Garea, Á. Ordóñez, D. B. Heras, and F. Argüello, "HypeRvieW: an open source desktop application for hyperspectral remote-sensing data processing," *International Journal of Remote Sensing*, vol. 37, no. 23, pp. 5533–5550, 2016

– A. S. Garea, D. B. Heras, and F. Arguëllo, "An open source desktop application for classification of remote sensing data," in *2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 1. IEEE, 2015, pp. 316–321

1.3 Methods and tools

Provided the complexity of the classification schemes proposed in this thesis and the large amount of data to be processed, the efficient computation of the proposals is a requirement. C/C++ was the base language for the implementations, OpenMP was selected for exploiting the multicore architectures available in Personal Computers (PCs), and CUDA was selected for exploiting the GPU, as explained in Section 1.3.1. In all cases different efficient libraries were used for specific computations.

The experiments were executed under Linux using the GNU Compiler Collection (GCC) for the CPU implementations and the NVIDIA CUDA Compiler (NVCC) for the CUDA implementations, with full optimization flags -O3 in both cases. The main specifications of the CPU and GPU used in our experiments are explained in Section 1.3.2. Regarding the evaluation of the proposed methods, 1.3.4 explains the evaluation measures in terms of both accuracy of the classification and execution time that were used in this thesis.

The different datasets used in this work correspond to multi and hyperspectral images and were obtained by sensors at large or medium distances as described in Section 1.3.5. The experimental conditions for the experiments are also summarized in that section.

1.3.1 Parallel programming models

This section introduces the OpenMP [103] and CUDA [94] programming models. First, we will focus on OpenMP, an Application Programming Interface (API) developed for writing multi-threaded application on shared memory architectures. Then, we will introduce the CUDA platform designed by NVIDIA. It comprises a programming model and a parallel computing platform that enables the use of NVIDIA GPUs.

OpenMP

OpenMP is a standard and portable API for writing shared memory parallel programs [103]. It means that it is designed for systems were all their threads have direct access to all available

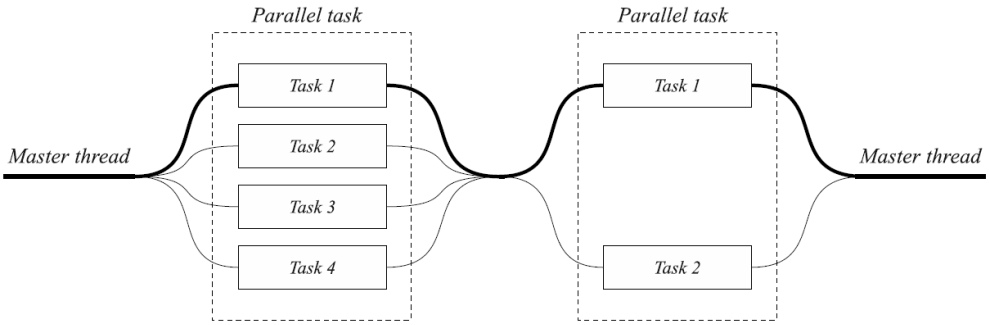


Figure 1.1: OpenMP fork-join parallel model. A program calls OpenMp directives that create a set of threads (fork), all of them coordinated by a master thread, to work in parallel. When all the threads finish their task, they are joined.

memory. OpenMP provides the programmer with high-level tools (compiler directives and library routines) that facilitate the parallelization of serial programs in Fortran and C/C++.

In the OpenMP execution model, a thread called *master thread* is always defined and it exists during the execution of the program. Only when a parallel region is encountered, additional threads are created to perform the parallel task. The master thread is responsible for creating and activating these threads, whose number can be defined by the user or by the system where the program is being executed. The synchronization of all the threads takes place at the end of the parallel region with an implicit barrier. Once the last thread has finished its work and, therefore, it has reached to the barrier, the execution continues with the master thread until the a new parallel region is encountered.

The shared memory model used by OpenMP allows each thread, inside a parallel region, to access both shared and private memory. All threads have read and write access to variables stored into the shared memory. Furthermore, a thread may access, or modify, variables in its own private memory, but not in the private memory of a different thread. The lifetime of a private variable is restricted and it only exists within a parallel region. Figure 1.1 shows how an OpenMP program runs as defined by the fork-join model.

The OpenMP API allows to set up several work-sharing scheduling strategies (static, dynamic, guided, auto, etc), by defining different chunk sizes (number of iterations inside a loop corresponding to a thread), resulting in different data access patters. An optimal selection of the scheduling strategy parameters maximizes the execution efficiency and avoids the

appearance of some performance issues such as *false sharing*, that would degrade the final efficiency. False sharing appears on systems where each processor has its own local cache when threads belonging to different processors modify variables that are stored on the same cache line. Selecting a chunck size that is a multiple of the cache line size will prevent cache lines being shared among different threads and, as a consequence, false sharing.

CUDA

CUDA is a general purpose parallel computing platform and programming model introduced by NVIDIA in 2006 [94]. This hardware/software platform combination enables NVIDIA GPUs to solve many complex computational problems in a more efficient way than on a CPU due to their large number of cores. A GPU is based on a series of Streaming Multiprocessors (SMs), each one containing many CUDA cores, shared memory, cache, etc. A SM architecture example is shown in Figure 1.2. From the NVIDIA G8x series [105] onwards, all the NVIDIA GPUs support CUDA. One of the first, the NVIDIA's G80 series, included a total of 16 SMs, each one with 8 CUDA cores and a global memory up to 1.5 GB. These figures are dwarfed by NVIDIA's latest GPU releases like the GeForce RTX 30 series [106] with up to 10496 CUDA cores and a maximum of 24 GB of global memory. But regardless of the improvements made thanks to advances in technology, the overall design is based on the same idea. That is, each multiprocessor is designed to execute threads concurrently, using a Single Instruction, Multiple Thread (SIMT) parallel programming model. This model combines the Single Instruction, Multiple Data (SIMD) vector processing model characteristics with multithreading.

The way NVIDIA GPUs execute programs is by invoking parallel functions called kernels that run as one-dimensional, two-dimensional or three-dimensional grid of blocks of threads. Each block is assigned to any of the available SM and so, not all blocks run concurrently. Since the communication between threads belonging to different blocks is not possible, the execution of the blocks must be independent. Figure 1.3 shows a two-dimensional grid of blocks each one with a configuration of 4×4 threads.

GPUs can manage thousands of threads that execute the same instruction over different data by grouping them into sets of 32 threads called wraps. The wrap is the minimum size of collaborative unit and it is hardware defined. To define the feature set (both hardware and software) of each device, NVIDIA uses a version number (1.x, 2.x, etc) called Compute Capability (CC). This number gives technical specifications such as: the maximum number

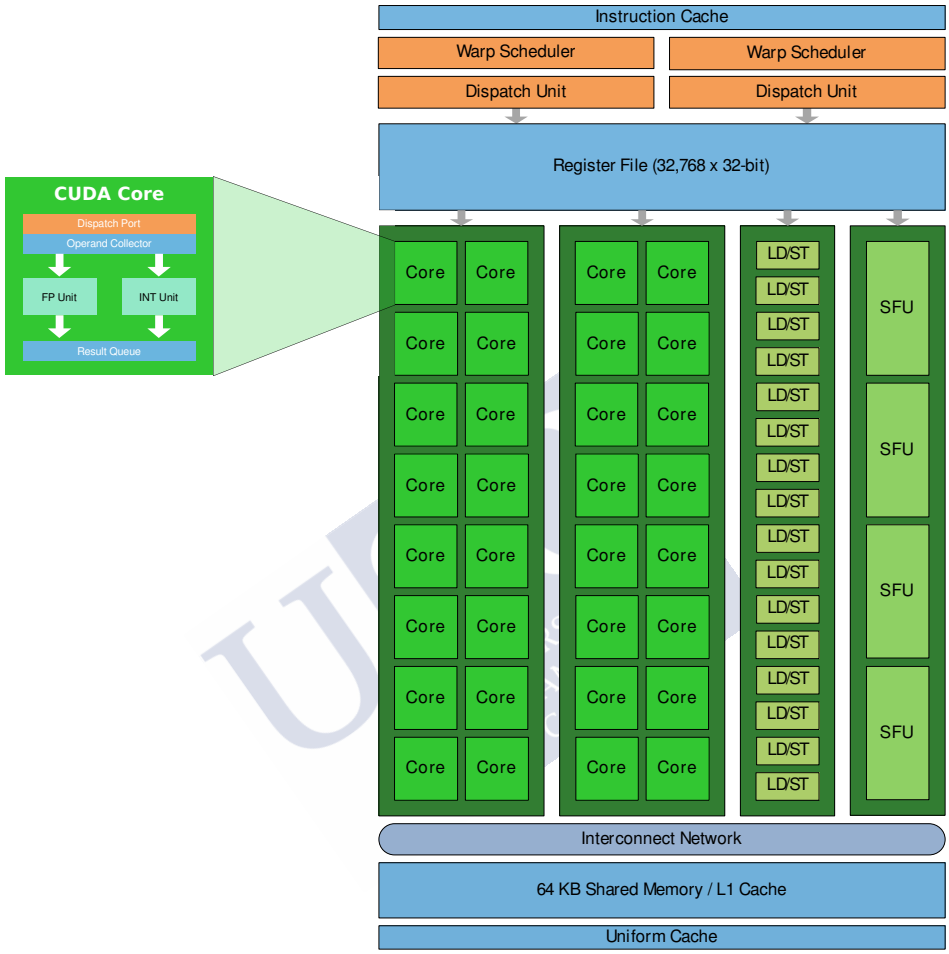


Figure 1.2: SM architecture for a GPU of the Fermi family [104].

of threads per block, the maximum number of blocks per grid or the amount of local memory per thread. The latest GPU available, GeForce RTX 30 Series [106], has a CC of 8.6.

The memory hierarchy is an important aspect to consider for a good performance [107]. Initially, the GPU and CPU had their own memory spaces in their respective Dynamic Random Access Memories (DRAMs), and the programmer was responsible for moving data between the CPU and GPU before and after a kernel was executed. In CUDA 6.0 the concept of unified memory was introduced for the first time. It defines a managed memory space in

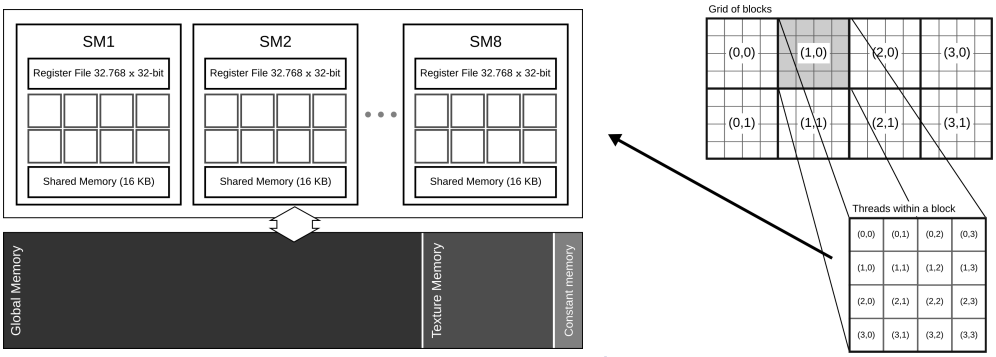


Figure 1.3: Grid of blocks and block of threads scheduled to any of the available SMs

which all processors (CPU and GPU) see a single coherent memory image with a common address space.

As important as choosing a good configuration of the pool of threads is to make them access memory in an efficient way. Certain memory access patterns enable the hardware to coalesce groups of reads or writes of multiple data items into one operation. Data that cannot be laid out so as to enable coalescing, or that doesn't have enough locality to use the L1 cache effectively, will tend to see lower speedups when used in computations on GPUs. Thus, there should be some coherence in memory access by adjacent threads running on the device to reduce the number of accesses to global memory [108].

1.3.2 Hardware

In this thesis, several low-cost computing infrastructures such as multicore CPUs and GPUs have been used with the aim of providing solutions to be executed by a standard remote sensing user. So, the proposed schemes can be executed efficiently without requiring the use of more sophisticated equipment such as supercomputers. Table 1.1 summarizes the main specifications of the different PCs used. In addition, two different General-Purpose Graphic Processing Units (GPGPUs) architectures have also been used in the course of this work to evaluate the proposed schemes: a NVIDIA GeForce GTX TITAN (Kepler [109]) and a NVIDIA GeForce GTX 1070 (Pascal [110]). Table 1.2 shows the GPU model, the CC of the graphic cards, available resources, and CUDA versions used in this thesis.

Table 1.1: CPU hardware specification.

CPU hardware	**# of cores**	**Core clock (MHz)**	**RAM (GB)**	**L1 (KB)**	**L2 (KB)**	**L3 (KB)**
Intel Core2 Quad Q9450	4	2667	6	32	6	-
Intel Xeon E5-2609v2	4	2500	16	32	256	10
Intel i5-6600	4	3300	32	32	256	6

Table 1.2: GPU model, CC, resources and CUDA version of the GPUs used in this thesis

	GTX TITAN (Kepler)	**GTX 1070 (Pascal)**
CC	3.5	6.1
SMs	14	15
CUDA cores per SM	192	128
Threads per block	1024	1024
Total threads per SM	2048	2048
Number of blocks per SM	16	32
Active warps per SM	64	64
Registers per SM	65536	65536
Registers per thread	255	255
Device memory	6 GB	8 GB
Shared memory	48 KB	96 KB
L1 cache	48 KB	48 KB
L2 cache	1536 KB	2048 KB
CUDA version	7.5	8.0

1.3.3 Software

Most of the codes developed in this thesis were coded using C/C++ language and they were built using GCC [111]. In some cases, the Python [112] language was also used. For the GPU implementations, the compiler was NVCC. OpenMp was considered for the multicore version of the codes. In the cases where Matlab was used, the version was 2015B.

As many efficient libraries are available for the different languages and platforms, several optimized mathematical libraries were exploited. Libraries such as CUBLAS [113] and CULA [114] were applied to the GPU codes while, for the CPU implementation, the mathematical optimized library called OpenBLAS [115] was used. These libraries allowed a more efficient computation of the PCA in the EMP algorithms that extract the main features of the images. Regarding the implementation of schemes based on DL, Python-based libraries such as TensorFlow [90] and Pytorch [91] were selected. In addition, Caffe [39], a DL framework for training and deploying general-purpose CNNs and other deep models efficiently on commodity architectures, and CUDA Deep Neural Network (cuDNN) [116], a GPU-accelerated library of primitives for deep neural networks, were used to code a classification scheme based on CNNs.

Finally, the input/output operations were optimized by using libraries such as GDAL [117], a translator library for raster geospatial data formats used to read multispectral Tag Image File Format (TIFF) images, and OpenCV [118], a computer vision software library, used to graphically display some results.

1.3.4 Performance measures

The following paragraphs describe the different performance measures that have been used to evaluate the classification schemes presented in this thesis. These measures evaluate classification accuracy and computational efficiency.

The process of classifying a hyperspectral image consists in assigning a label to each pixel, resulting in a classification map. The accuracy of a classification is based on the comparison of the classification map obtained and the available reference map. The construction of an accurate reference map, also known as ground-truth, is in many cases a long-term process involving different agents. For example, for the case of the multispectral images used in this thesis, information from vegetation inventories, field visits and the expertise of the forestry

experts along with the analysis of canopy textures were the main elements considered in producing the reference data.

In order to quantitatively evaluate the classification accuracies, we have used the Class-Specific (CS) accuracy, the Overall Accuracy (OA), the Average Accuracy (AA), the Kappa coefficient of agreement (k) [119], the standard McNemar test [120, 121], the Quantity Disagreement (QD) [122], the Allocation Disagreement (AD) [122], and the F1-score [123] as the criteria for assessing the accuracy of the proposal classification schemes.

More formally, let C_i represent the class i, C_{ij} the number of pixels classified as class j and referenced as belonging to class i, and K the number of classes.

- CS (or producer's accuracy) is the percentage of correctly classified pixels for a given class i:

$$CS_i = \frac{C_{ii}}{\sum_j^K C_{ij}} \times 100\% . \tag{1.1}$$

- OA represents the total percentage of correctly classified pixels:

$$OA = \frac{\sum_i^K C_{ii}}{\sum_{ij}^K C_{ij}} \times 100\% . \tag{1.2}$$

- AA is the the average percentage of correctly classified pixels for each class. It is the mean of all the class-specific accuracies.:

$$AA = \frac{\sum_i^K CS_i}{K} \times 100\% . \tag{1.3}$$

- k is defined as the percentage of agreement corrected by the amount of agreement that could be expected due to chance alone, which ranges from 0 to 1. A value of k below 0.6 is interpreted as a moderate agreement, whereas a value above 0.8 can be understood as an almost perfect agreement.

- The McNemar test verifies the statistical significance of accuracy differences among pairs of schemes. It is considered to be significantly different at 95% confidence level with values greater than 1.96, and at 99% confidence level with values greater than 2.58. A positive sign indicates that the first classifier outperforms the second one.

- QD measures the disagreement between a classification map and the reference data in terms of proportion of the classes.

- AD measures the disagreement between a classification map and the reference data in terms of spatial allocation of the classes.
- The F1-score measures the accuracy of a test. It is computed using the harmonic mean from the precision and the recall of the test, where the precision of a class is the number of correctly predicted pixels for this class divided by the number of all the pixels identified as this class, including those not identified correctly, and the recall of a class is the number of correctly predicted pixels of the class divided by the number of all samples that should have been identified as pixels belonging to this class.

The schemes proposed in this thesis were also evaluated in terms of execution time. The GPU implementations were compared against both CPU and CPU multi-thread OpenMP implementations to obtain the speedup measure (S). S is defined as the improvement in execution time between two different implementations and computed as the ratio between the execution time of those implementations. The execution times obtained for the different schemes in this thesis represent the elapsed time from the start to the end of the computation, without taking into account file access times (read and write operations). The reason to exclude this input/output access time is that each scheme comprises different stages concatenated in a pipeline processing where the output of one stage is stored in memory to be used as input to the next stage.

Table 1.4: Main features of the datasets used in this thesis.

Dataset name	**Dimensions (V×H×B)**	**Size (MB)**
Pavia University	610×340×103	85.4
Indian Pines	145×145×220	18.5
Salinas	512×217×204	90.7
Pavia Centre	1096×715×102	319.7
Oitavén River	6689×6722×5	899.3
Ermidas Creek	11924×18972×5	4314.8
Xesta Basin	15424×10179×5	2994.5
Eiras Dam	5176×18221×5	1798.8
Pavia City	512×1400×102	292.5
Indiana	2678×614×220	1379.9

Table 1.3: Main characteristics of both the Reflective Optics System Imaging Spectrometer (ROSIS-03) and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensors.

Characteristic	**ROSIS-03**	**AVIRIS**
Angular Field of View (FOV)	16°	30°
Instantaneous Field of View (IFOV)	0.56 mrad	0.95 mrad
Number of pixels per line	512	614
Scan principle	Pushbroom	Whiskbroom
Ground resolution	1 m–6 m	20 m
Radiometric resolution	14 bits	10 bits
Spectral range	430 nm–860 nm	400 nm–2500 nm
Spectral sampling	4 nm	9.6 nm–10.0 nm
Inflight calibration	0.2 nm	0.5 nm

1.3.5 Datasets

The schemes proposed in this thesis were evaluated over hyperspectral and multispectral remote sensing images. The hyperspectral images were acquired by two different satellite sensors: ROSIS-03 and AVIRIS. In the case of the multispectral images, they were captured by the MicaSense RedEdge multispectral camera mounted on a custom UAV property of the Babcock company. The main specifications of the satellite sensors are detailed in Table 1.3 [124, 125, 126, 127, 128, 129].

The MicaSense RedEdge multispectral camera consists of five discrete sensors covering spectral channels at wavelengths of 475 nm (Blue), 560 nm (Green), 668 nm (Red), 717 nm (Edge), and 840 nm (NIR). Its spatial resolution is 8.2 cm/pixel at a height of 120 m with a horizontal field of view of 47.2° and a radiometric resolution of 11 bits.

Table 1.4 summarizes the names, dimensions (vertical $\times$ horizontal $\times$ bands), and size of the datasets in MB.

Pavia University dataset

This dataset was acquired by the ROSIS-03 hyperspectral sensor, covering the spectrum between 430 and 860 nm in 115 spectral bands (the twelve noisiest bands of this dataset were

removed so the final dataset has 103 spectral bands), corresponding mainly to the visible spectrum. This image is an urban area surrounding the University of Pavia [130], northern Italy. It has a spatial size of 610×340 pixels with a spatial resolution of 1.3 m per pixel, covering about 0.35 km^2. The number of classes in the reference data is nine: asphalt, meadows, gravel, trees, metal sheets, bare soil, bitumen, bricks, and shadows. The hyperspectral image and its reference data (ground truth) are shown in Figure 1.4. The number of available pixels per class in the reference data is shown in Table 1.5.

In order to show the similarity/dissimilarity among classes, the spectral means of all available classes in the ground-truth are displayed in Figure 1.5. We can observe how the tree and metal signatures are significantly different from the rest, while others such as bitumen and asphalt or gravel and bricks are more similar. The number of training pixels, known as training samples, used for the experiments is the same as in the reference works in the literature [131] in order to perform a reliable comparison. Details are shown in Table 1.6.

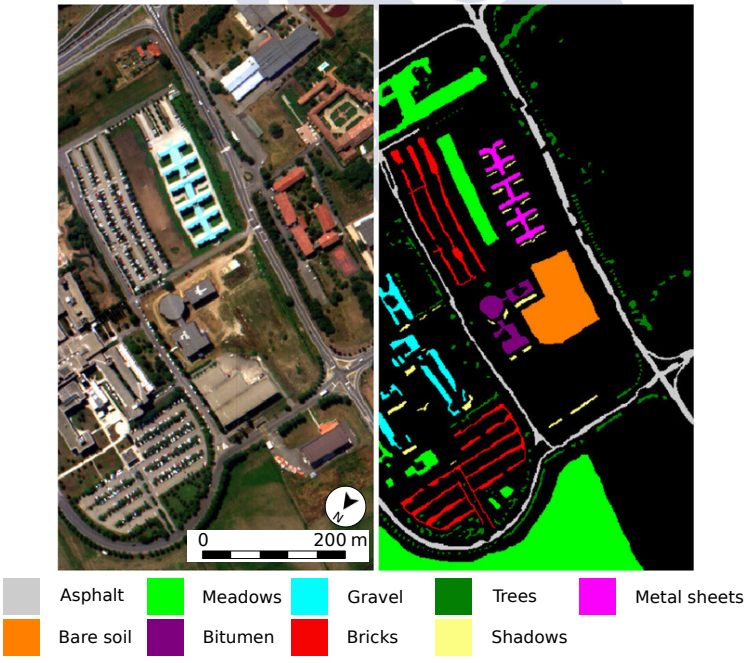


Figure 1.4: False color composite image, reference data and label meaning for the Pavia University dataset.

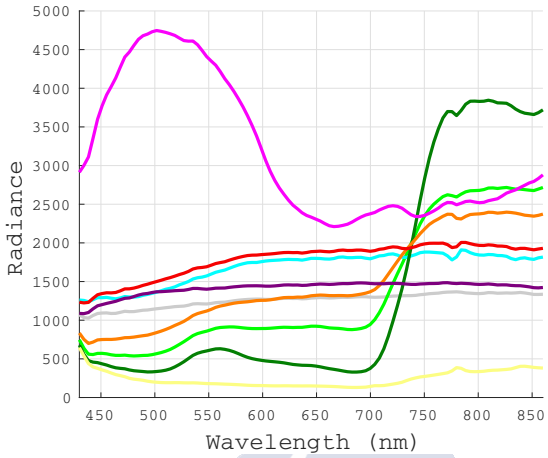


Figure 1.5: Spectral mean values for all the classes in the Pavia University dataset. The color code is the same as for Figure 1.4.

Table 1.5: Number of pixels per class for the Pavia University dataset.

Label	Name	# of available pixels
1	Asphalt	6631
2	Meadows	18649
3	Gravel	2099
4	Trees	3064
5	Metal sheets	1345
6	Base soil	5029
7	Bitumen	1330
8	Bricks	3682
9	Shadows	947
Total		42776

Table 1.6: Training and test samples for the Pavia University dataset.

Classes	Train	%	Test	%
Asphalt	548	8.26	6083	91.74
Meadows	540	2.90	18109	97.10
Gravel	392	18.68	1707	81.32
Trees	524	17.10	2540	82.90
Metal sheets	265	19.70	1080	80.30
Base soil	532	10.58	4497	89.42
Bitumen	375	28.20	955	71.80
Bricks	514	13.96	3168	86.04
Shadows	231	24.39	716	75.61
Total	3921	9.17	38855	90.83

Indian Pines dataset

This dataset was acquired by the AVIRIS hyperspectral sensor over the Indian Pine Test Site 3 (2×2 mile portion of NW Tippecanoe County, Indiana) on 12 June 1992 [132]. The image is a rural area that includes agricultural and forest zones. It has a spatial dimension of 145×145 pixels with a spatial resolution of 20 m and covers about 10 km^2. The spectral resolution is of 220 bands covering the spectral range from 400nm to 2500 nm, corresponding to the visible and infrared spectrum. The hyperspectral image and its reference data (ground truth) are shown in Figure 1.6. Two-thirds of the scene corresponds to agricultural areas and the rest corresponds to forest or other natural perennial vegetation. The number of classes in the reference data is sixteen and they are listed in Table 1.7.

The spectral means of all available classes in the ground-truth are displayed in Figure 1.7. As we can see, all the classes have a similar signature because they all correspond to vegetation species. Finally, the number of training samples for the experiments was selected in order to perform a reliable comparison to the reference works in the literature [131]. The detailed number of samples per class are shown in Table 1.8.

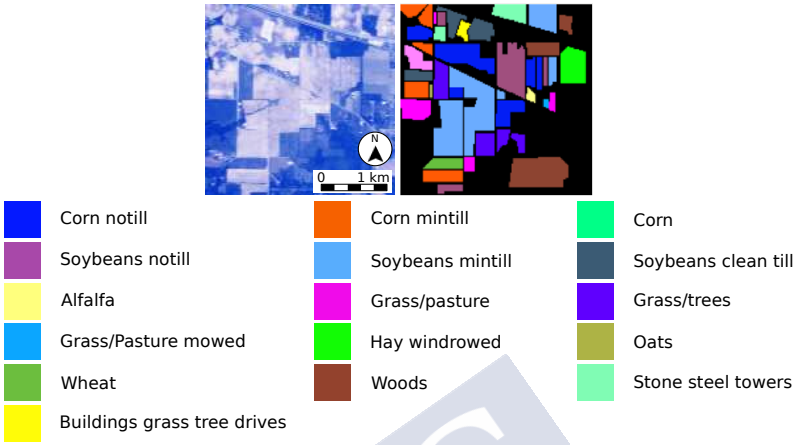


Figure 1.6: False color composite image, reference data and label meaning for the Indian Pines dataset.

Table 1.7: Number of pixels per class for the Indian Pines Dataset.

Label	Name	# of available pixels
1	Corn-no till	1434
2	Corn-min till	834
3	Corn	234
4	Soybeans-no till	968
5	Soybeans-min till	2468
6	Soybeans-clean till	614
7	Alfalfa	54
8	Grass/pasture	497
9	Grass/trees	747
10	Grass/pasture-mowed	26
11	Hay-windrowed	489
12	Oats	20
13	Wheat	212
14	Woods	1294
15	Bldg-grass-tree-drives	380
16	Stone-steel towers	95
Total		10366

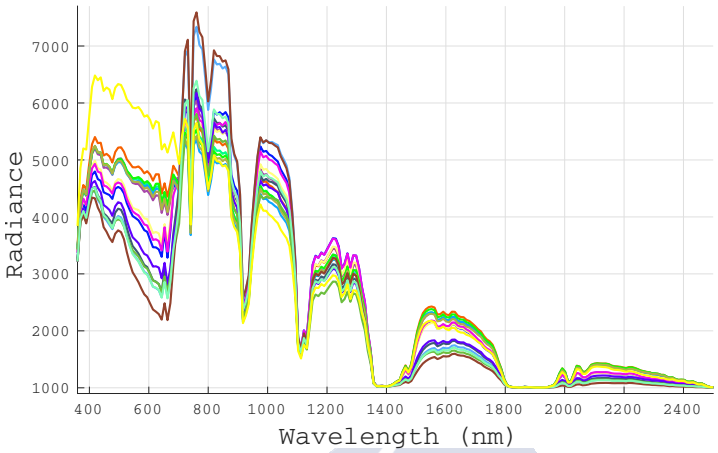


Figure 1.7: Spectral mean values for all the classes in the Indian Pines dataset. The color code is the same as for Figure 1.6.

Table 1.8: Training and test samples for the Indian Pines Dataset.

Classes	Train	%	Test	%
Corn-no till	50	3.49	1384	96.51
Corn-min till	50	6.00	784	94.00
Corn	50	21.37	184	78.63
Soybeans-no till	50	5.17	918	94.83
Soybeans-min till	50	2.03	2418	97.97
Soybeans-clean till	50	8.14	564	91.86
Alfalfa	15	27.78	39	72.22
Grass/pasture	50	10.06	447	89.94
Grass/trees	50	6.69	697	93.31
Grass/pasture-mowed	15	57.69	11	42.31
Hay-windrowed	50	10.22	439	89.78
Oats	15	75.00	5	25.00
Wheat	50	23.58	162	76.42
Woods	50	3.86	1244	96.14
Bldg-grass-tree-drives	50	13.16	330	86.84
Stone-steel towers	50	52.63	45	47.37
Total	695	6.70	9671	93.30

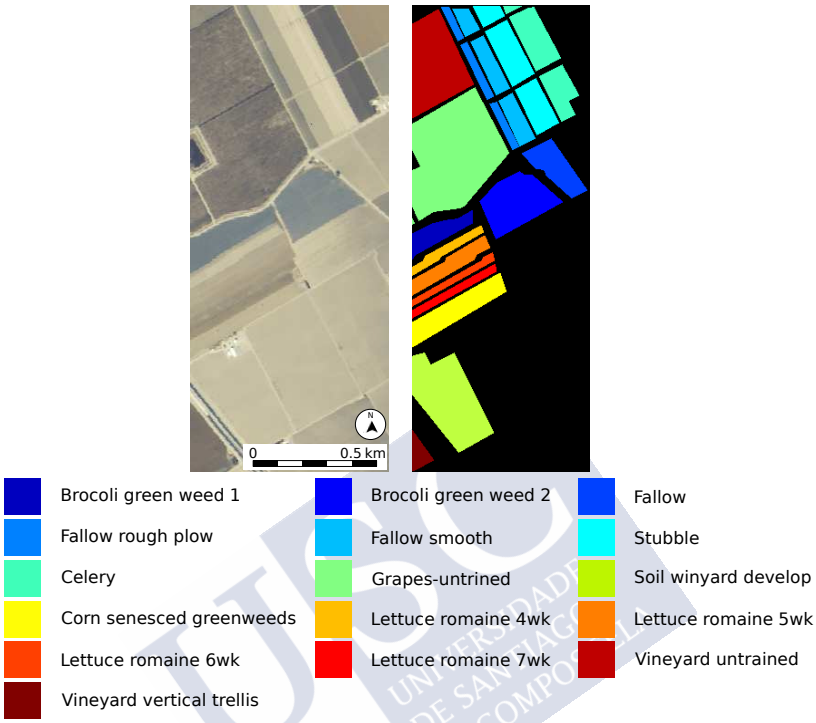


Figure 1.8: False color composite image, reference data and label meaning for the Salinas dataset.

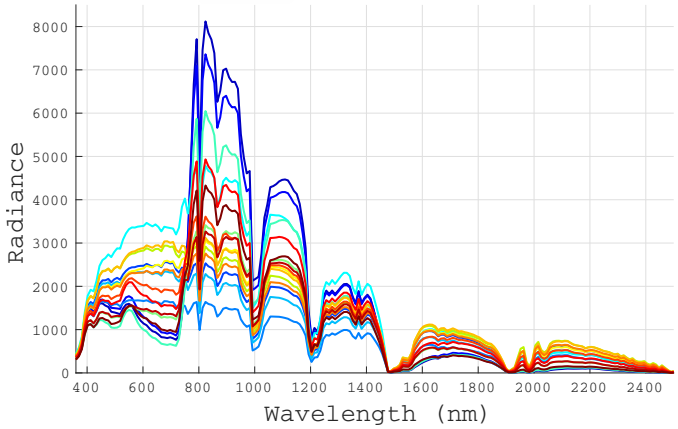


Figure 1.9: Spectral mean values for all the classes in the Salinas dataset. The color code is the same as for Figure 1.8.

Salinas dataset

This dataset was acquired by the AVIRIS hyperspectral sensor covering the spectrum between 400nm and 2500nm in 204 spectral bands (after discarding twenty water absorption spectral bands) over the Salinas Valley [133], California. It has a spatial dimension of 512×217 pixels with a spatial resolution of 3.7 m per pixel, covering about 1.5 km^2. The hyperspectral image and its reference data (ground-truth) are shown in Figure 1.8. The number of classes in the reference data is sixteen, including vegetables, bare soil, and vineyard fields. They are shown in Table 1.9.

The spectral means of all classes available in the ground-truth are displayed in Figure 1.9. Similar to Indian Pines dataset, all the classes correspond to vegetation species and therefore their signatures are quite alike. Finally, the number of training samples for the experiments is 2% of the available samples. This percentage was selected in order to perform a reliable comparison to the reference works in the literature [134]. The number of samples per class are detailed in Table 1.10.

Table 1.9: Number of pixels per class for the Salinas Dataset.

Label	Name	# of available pixels
1	Brocoli-green-weeds-1	2009
2	Brocoli-green-weeds-2	3726
3	Fallow	1976
4	Fallow-rough-plow	1394
5	Fallow-smooth	2678
6	Stubble	3959
7	Celery	3579
8	Grapes-untrained	11271
9	Soil-vinyard-develop	6203
10	Corn-senesced-green-weeds	3278
11	Lettuce-romaine-4wk	1068
12	Lettuce-romaine-5wk	1927
13	Lettuce-romaine-6wk	916
14	Lettuce-romaine-7w	1070
15	Vinyard-untrained	7268
16	Vinyard-vertical-trellis	1807
Total		54129

Table 1.10: Training and test samples for the Salinas Dataset.

Classes	Train	%	Test	%
Brocoli-green-weeds-1	40	1.99	1969	98.01
Brocoli-green-weeds-2	74	1.99	3652	98.01
Fallow	39	1.97	1937	98.03
Fallow-rough-plow	27	1.94	1367	98.06
Fallow-smooth	53	1.98	2625	98.02
Stubble	79	2.00	3880	98.00
Celery	71	1.98	3508	98.02
Grapes-untrained	225	2.00	11046	98.00
Soil-vinyard-develop	124	2.00	6079	98.00
Corn-senesced-green-weeds	65	1.98	3213	98.02
Lettuce-romaine-4wk	21	1.97	1047	98.03
Lettuce-romaine-5wk	38	1.97	1889	98.03
Lettuce-romaine-6wk	18	1.97	898	98.03
Lettuce-romaine-7w	21	1.96	1049	98.04
Vinyard-untrained	145	2.00	7123	98.00
Vinyard-vertical-trellis	36	1.99	1771	98.01
Total	1076	1.99	53053	98.01

Pavia Centre dataset

This dataset was acquired by the ROSIS-03 hyperspectral sensor over the City of Pavia [135], northern Italy. It has a spatial size of 1096×715 pixels with a resolution of 1.3 m, and covers about 2 km^2. The spectral resolution is 102 spectral bands, after removing thirteen noisy channels, covering the spectral range from 430 and 860 nm. The hyperspectral image and its reference data (ground-truth) are shown in Figure 1.10. The number of classes in the reference data is nine: water, trees, meadows, bricks, bare soil, asphalt, bitumen, tile, and shadows. The number of available pixels by class in the reference data is shown in Table 1.11.

The spectral means of all available classes in the ground-truth are displayed in Figure 1.11. The number of training samples for the experiments is the same as in the reference works in the literature [131] in order to perform a reliable comparison. The values are shown in Table 1.12.

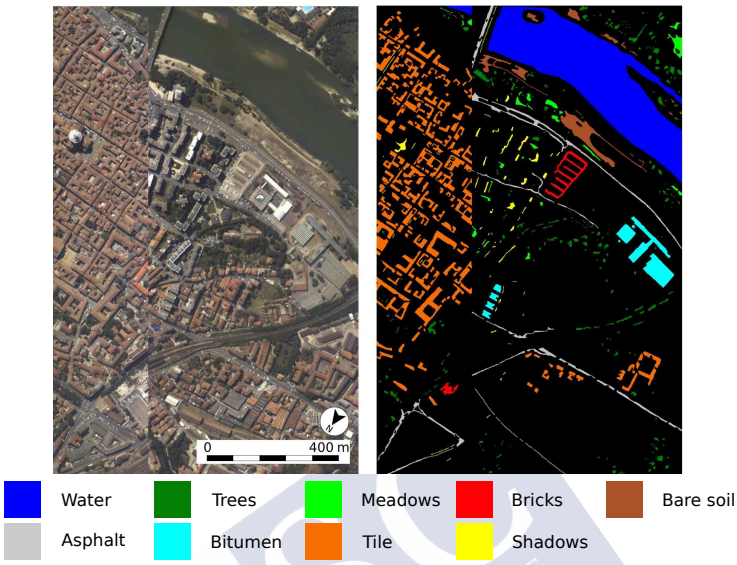


Figure 1.10: False color composite image, reference data and label meaning for the Pavia Centre dataset.

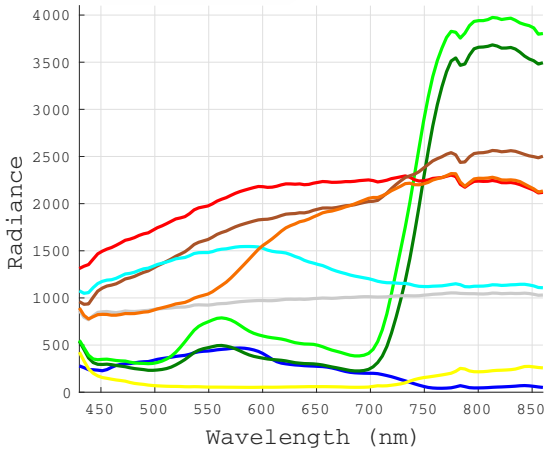


Figure 1.11: Spectral mean values for all the classes in the Pavia Centre dataset. The color code is the same as for Figure 1.10.

Table 1.11: Number of pixels per class for the Pavia Centre dataset.

Label	Name	# of available pixels
1	Water	65971
2	Tree	7598
3	Meadow	3090
4	Bricks	2685
5	Bare Soil	6584
6	Asphalt	9248
7	Bitumen	7287
8	Tile	42826
9	Shadow	2863
Total		148152

Table 1.12: Training and test samples for the Pavia Centre dataset.

Classes	Train	%	Test	%
Water	824	1.25	65147	98.75
Tree	820	10.79	6778	89.21
Meadow	824	26.67	2266	73.33
Bricks	808	30.09	1877	69.91
Bare Soil	820	12.45	5764	87.55
Asphalt	816	8.82	8432	91.18
Bitumen	808	11.09	6479	88.91
Tile	1260	2.94	41566	97.06
Shadow	476	16.63	2387	83.37
Total	7456	5.03	140696	94.97

Oitavén River dataset

This dataset was acquired on July 18, 2018 by the MicaSense RedEdge multispectral camera covering the spectrum between 475 and 840 nm in 5 spectral bands at a height of 120 meters over the watershed of the Oitavén River on its way through the local village of A Ponte. The size of the image is 6689 × 6722 pixels. The color composite of this multispectral image and its reference data (ground-truth) are shown in Figure 1.12. It reveals some misregistered areas in the lower right part of the scene, probably caused by the unpredictable light variations along the flight. Leaving aside those artifacts, the image is indeed in very good shape. In it, several tree species (oak and eucalyptus are specially abundant) and a number of artificial structures

can be spotted. The total number of classes in the reference data is eleven: water, oak, tiles, meadows, asphalt, bare soil, rock, concrete, autochthonous vegetation, eucalyptus, and pines. The number of available pixels per class in the reference data is show in Table 1.13.

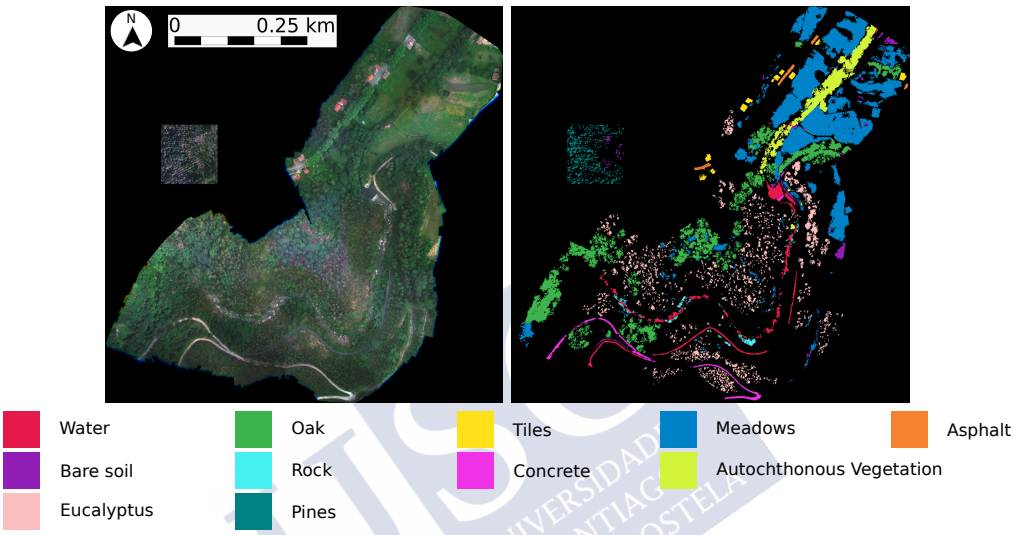


Figure 1.12: False color composite image, reference data and label meaning for the Oitavén River dataset.

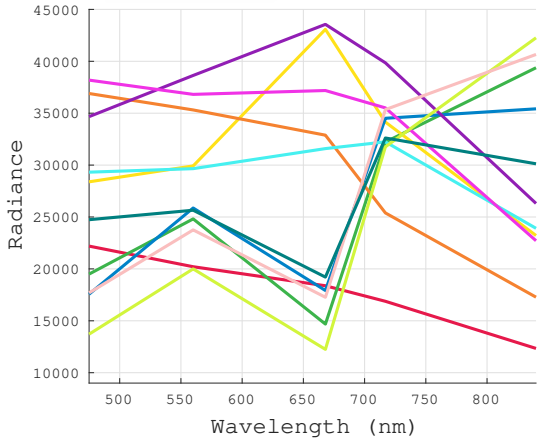


Figure 1.13: Spectral mean values for all the classes in the Oitavén River dataset. The color code is the same as for Figure 1.12.

The spectral means of all available classes in the reference data are displayed in Figure 1.13. We can see that the spectral signatures corresponding to the vegetation species follow a characteristic pattern of this type of class that clearly differentiates them from the rest. The number of training samples and segments for the experiments are shown in Table 1.14.

Table 1.13: Number of pixels per class for the Oitavén River dataset.

Label	Name	# of available pixels
1	Water	309248
2	Oak	1374889
3	Tiles	78785
4	Meadows	2440331
5	Asphalt	43861
6	Bare Soil	113329
7	Rock	79152
8	Concrete	128022
9	Autochthonous Vegetation	458565
10	Eucalyptus	863698
11	Pines	193884
Total		6083764

Table 1.14: Training and test samples for the Oitavén River dataset.

	Pixels				Segments			
Classes	Train	%	Test	%	Train	%	Test	%
Water	52827	17.08	256421	82.92	96	14.93	547	85.07
Oak	209344	15.23	1165545	84.77	554	14.99	3142	85.01
Tiles	12548	15.93	66237	84.07	27	14.52	159	85.48
Meadows	368637	15.11	2071694	84.89	903	14.99	5122	85.01
Asphalt	6977	15.91	36884	84.09	14	14.74	81	85.26
Bare Soil	16841	14.86	96488	85.14	43	14.98	244	85.02
Rock	10829	13.68	68323	86.32	32	14.68	186	85.32
Concrete	20316	15.87	107706	84.13	55	14.91	314	85.09
Autochthonous Vegetation	70758	15.43	387807	84.57	190	14.93	1083	85.07
Eucalyptus	121074	14.02	742624	85.98	402	14.98	2282	85.02
Pines	26386	13.61	167498	86.39	68	14.85	390	85.15
Total	916537	15.07	5167227	84.93	2384	14.96	13550	85.04

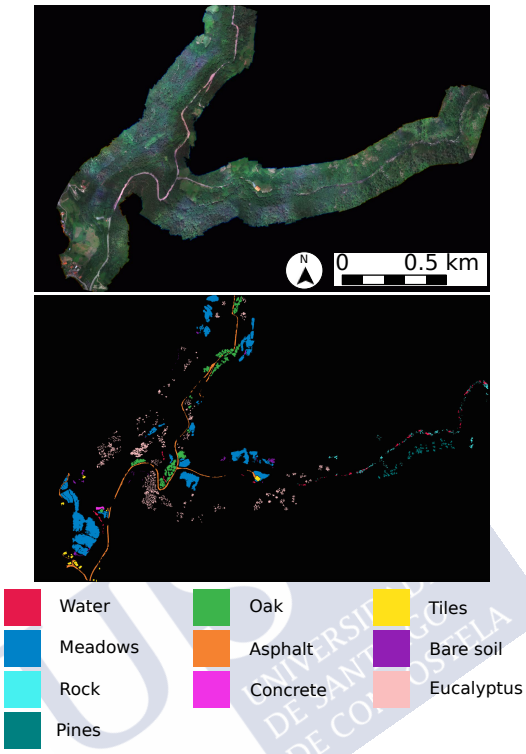


Figure 1.14: False color composite image, reference data and label meaning for the Ermidas Creek dataset.

Ermidas Creek dataset

This dataset was acquired on July 18, 2018 by the MicaSense RedEdge multispectral camera covering the spectrum between 475 and 840 nm in 5 spectral bands at a height of 120 meters. This image is located 500 meters east from the Oitavén River dataset, where the Ermidas Creek meets the Oitavén River. The flight covers the banks where these two water flows converge. The dimensions of this dataset are 11924 × 18972 pixels and present some shaded areas in the surroundings of the main road crossing the image from north to south. The color composite of this multispectral image and its reference data (ground-truth) are shown in Figure 1.14. In relation to the vegetation masses, there are plenty of oak and eucalyptus canopies on the west whereas the east side is sparsely populated by pines. The total number of classes in the reference data is ten: water, oak, tiles, meadows, asphalt, bare soil, rock,

concrete, eucalyptus, and pines. The number of available pixels per class in the reference data is show in Table 1.15.

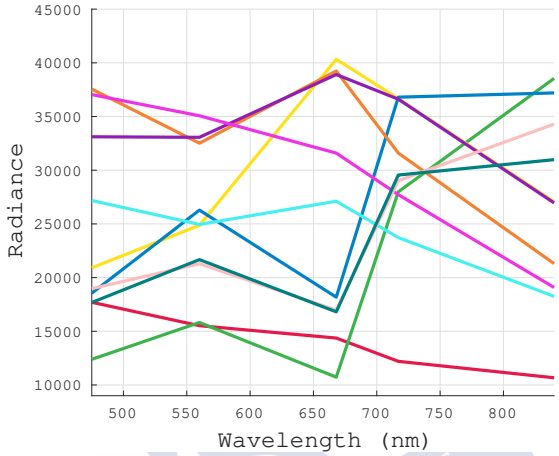


Figure 1.15: Spectral mean values for all the classes in the Ermidas Creek dataset. The color code is the same as for Figure 1.14.

The spectral means of all available classes in the reference data are displayed in Figure 1.15. The number of training samples and segments for the experiments are shown in Table 1.16.

Table 1.15: Number of pixels per class for the Ermidas Creek dataset.

Label	Name	# of available pixels
1	Water	163930
2	Oak	804040
3	Tiles	138678
4	Meadows	3423506
5	Asphalt	737409
6	Bare Soil	123416
7	Rock	174088
8	Concrete	32866
9	Autochthonous Vegetation	-
10	Eucalyptus	1135997
11	Pines	184547
Total		6918477

Table 1.16: Training and test samples for the Ermidas Creek dataset.

	Pixels				Segments			
Classes	Train	%	Test	%	Train	%	Test	%
Water	26435	16.13	137495	83.87	48	14.95	273	85.05
Oak	124614	15.50	679426	84.50	331	14.98	1878	85.02
Tiles	19911	14.36	118767	85.64	45	14.85	258	85.15
Meadows	530229	15.49	2893277	84.51	1273	15.00	7216	85.00
Asphalt	114028	15.46	623381	84.54	255	14.96	1449	85.04
Bare Soil	19353	15.68	104063	84.32	47	14.92	268	85.08
Rock	22732	13.06	151356	86.94	67	14.99	380	85.01
Concrete	4362	13.27	28504	86.73	12	15.00	68	85.00
Autochthonous Vegetation	-	-	-	-	-	-	-	-
Eucalyptus	168238	14.81	967759	85.19	484	14.98	2746	85.02
Pines	27742	15.03	156805	84.97	77	14.92	439	85.08
Total	1057644	15.29	5860833	84.71	2639	14.98	14975	85.02

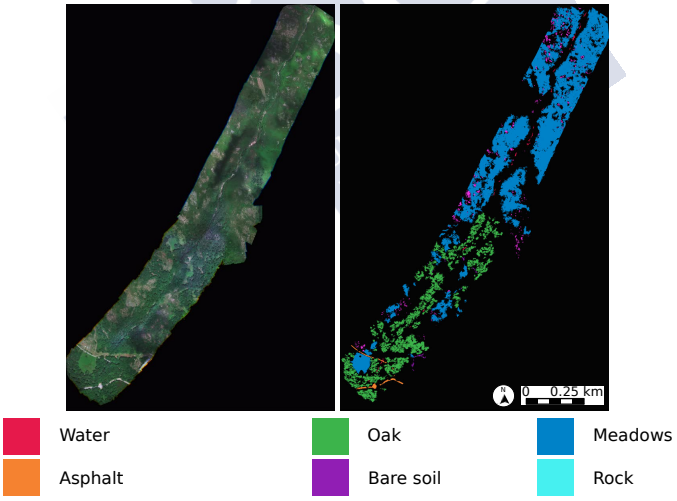


Figure 1.16: False color composite image, reference data and label meaning for the Xesta Basin dataset.

Xesta Basin dataset

This dataset was acquired near the border between the Galician provinces of Pontevedra and Ourense, where the source of the Xesta River is located, by the MicaSense RedEdge multispectral camera covering the spectrum between 475 and 840 nm in 5 spectral bands at a

height of 120 meters. This is an exquisitely preserved basin with vast fields of low vegetation, unique rocky areas as well as oak canopies. The image was captured on July 6, 2018. The dimensions of this dataset are 15424×10179. The composite color of this multispectral image and its reference data (ground-truth) are shown in Figure 1.16. The total number of classes in the reference data is six: water, oak, meadows, asphalt, bare soil, and rock. The number of available pixels by class in the reference data is show in Table 1.17.

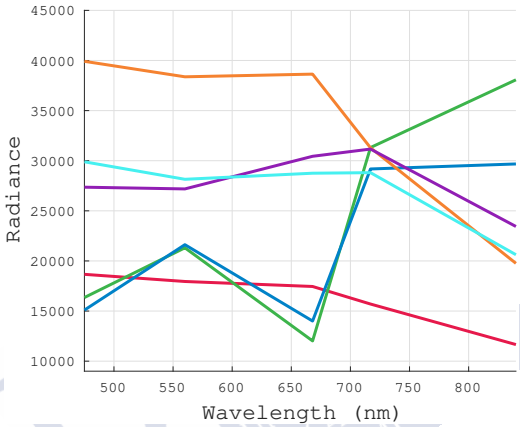


Figure 1.17: Spectral mean values for all the classes in the Xesta Basin dataset. The color code is the same as for Figure 1.16.

Table 1.17: Number of pixels per class for Xesta Basin dataset.

Label	Name	# of available pixels
1	Water	71756
2	Oak	3821501
3	Tiles	-
4	Meadows	12468609
5	Asphalt	130612
6	Bare Soil	49468
7	Rock	503776
8	Concrete	-
9	Autochthonous Vegetation	-
10	Eucalyptus	-
11	Pines	-
Total		17045722

The spectral means of all available classes in the reference data are displayed in Figure 1.17. We can see that within the spectral signatures belonging to the non-vegetation class, water and asphalt are clearly distinguishable, while rock and bare soil share more similarities. The same is true for the signatures of the vegetation classes, where oak and meadow are very similar. The number of training samples and training segments for the experiments are shown in Table 1.18.

Table 1.18: Training and test samples for Xesta Basin dataset.

	Pixels				Segments			
Classes	Train	%	Test	%	Train	%	Test	%
Water	10747	14.98	61009	85.02	20	13.7	126	86.3
Oak	444515	11.63	3376986	88.37	1103	11.17	8775	88.83
Tiles	-	-	-	-		-	-	-
Meadows	1370935	11	11097674	89	3377	10.82	27837	89.18
Asphalt	17017	13.03	113595	86.97	40	12.35	284	87.65
Bare Soil	9989	20.19	39479	79.81	25	20.49	97	79.51
Rock	50617	10.05	453159	89.95	151	11.04	1217	88.96
Concrete	-	-	-	-		-	-	-
Autochthonous Vegetation	-	-	-	-		-	-	-
Eucalyptus	-	-	-	-		-	-	-
Pines	-	-	-	-		-	-	-
Total	1903820	11.17	15141902	88.83	4716	10.95	38336	89.05

Eiras Dam dataset

This dataset was acquired on July 6, 2018 by the MicaSense RedEdge multispectral camera covering the spectrum between 475 and 840 nm in 5 spectral bands at a height of 120 meters over the Eiras Dam. The image dimensions are 5176 × 18221 pixels. The color composite of this multispectral image and its reference data (ground-truth) are shown in Figure 1.18. Finished in 1977, Eiras Dam is the reservoir that supplies running water to the city of Vigo. The dam outlet gives way to a river bed with many meanders, surrounded to the south by dense oak canopies and artificial paths. The lack of vegetation that can be noticed on the north banks was due to a wikdfire in 2018. To the east, the dam crest along the main penstock appears bounded by some sparse pines crowns. The total number of classes in the reference data is ten: water, oak, tiles, meadows, asphalt, bare soil, rock, concrete, eucalyptus, and pines. The number of available pixels per class in the reference data is shown in Table 1.19.

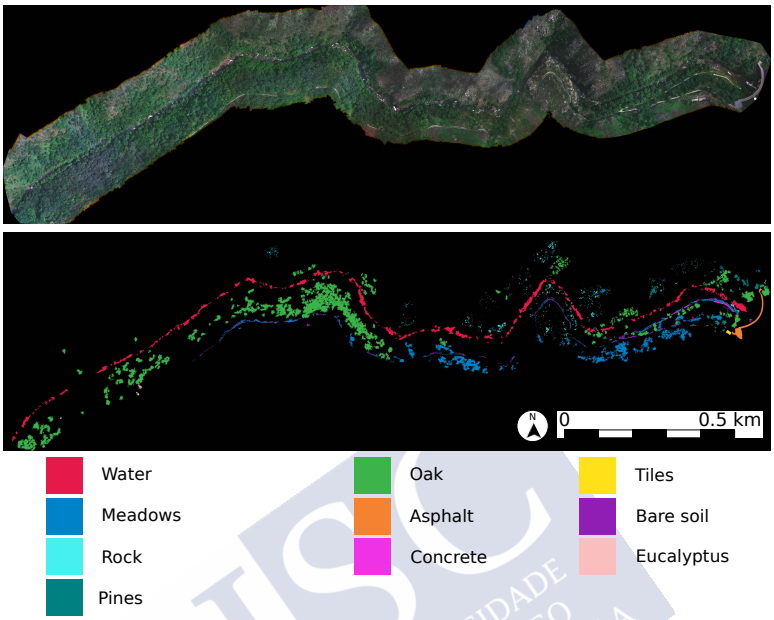


Figure 1.18: False color composite image, reference data and label meaning for the Eiras Dam dataset.

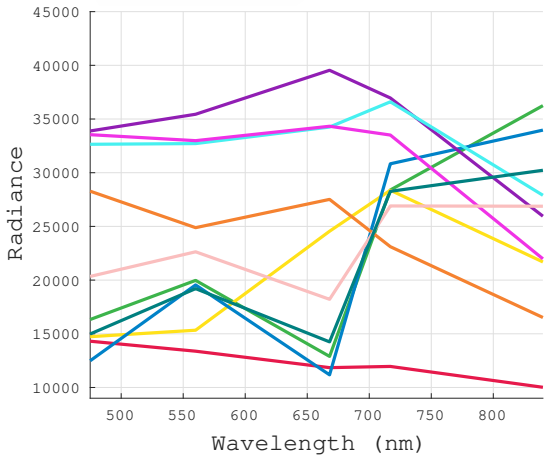


Figure 1.19: Spectral mean values for all the classes in the Eiras Dam dataset. The color code is the same as for Figure 1.18.

Table 1.19: Number of pixels per class for Eiras Dam dataset.

Label	Name	# of available pixels
1	Water	734617
2	Oak	2067380
3	Tiles	8232
4	Meadows	773964
5	Asphalt	85209
6	Bare Soil	96935
7	Rock	144800
8	Concrete	27061
9	Autochthonous Vegetation	-
10	Eucalyptus	8451
11	Pines	95132
Total		4041781

The spectral means of all available classes in the reference data are displayed in Figure 1.19. We can observe that the vegetation species have similar signatures which makes their identification difficult. The number of training samples and segments for the experiments are shown in Table 1.20.

Table 1.20: Training and test samples for Eiras Dam dataset.

	Pixels				Segments			
Classes	Train	%	Test	%	Train	%	Test	%
Water	113319	15.43	621298	84.57	214	17.15	1034	82.85
Oak	317454	15.36	1749926	84.64	859	14.99	4873	85.01
Tiles	3262	39.63	4970	60.37	8	40	12	60
Meadows	106951	13.82	667013	86.18	271	14.95	1542	85.05
Asphalt	13765	16.15	71444	83.85	30	16.57	151	83.43
Bare Soil	19448	20.06	77487	79.94	55	16.62	276	83.38
Rock	11018	7.61	133782	92.39	40	18.43	177	81.57
Concrete	9148	33.81	17913	66.19	25	29.76	59	70.24
Autochthonous Vegetation	-	-	-	-	-	-	-	-
Eucalyptus	2691	31.84	5760	68.16	8	33.33	16	66.67
Pines	19346	20.34	75786	79.66	55	20.75	210	79.25
Total	616402	15.25	3425379	84.75	1565	15.78	8350	84.22

Pavia City Dataset

This dataset was acquired by the ROSIS-03 hyperspectral sensor covering the spectrum between 430 and 860 nm in 115 spectral bands over the city of Pavia. It has a spatial size of 512×1400 pixels with a resolution of 1.3 m, and covers about 1 km^2. After removing noisy bands, the final spectral resolution of the dataset is 102 bands. Following the specifications of [86] two disjoint regions were used. The total number of classes for the full dataset is five, but because one of the classes has no samples in both disjoint regions, only four of the available classes were used: roads, vegetation, shadows, and buildings. The hyperspectral image and its reference data (ground-truth) are shown in Figure 1.20. Table 1.21 shows the number of samples in the full image for the selected classes. The source region, shown in

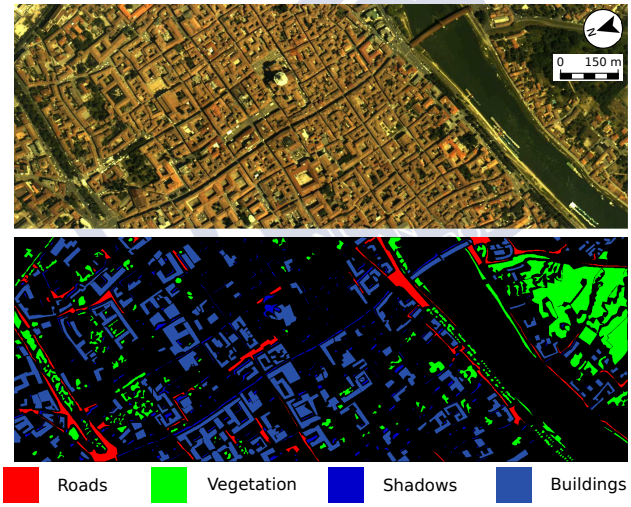


Figure 1.20: False color composite image, reference data and label meaning for the Pavia City dataset.

Figure 1.21, has a size of 124×173 pixels where its upper-left corner corresponds to the coordinates (907, 266) in the Pavia City image and its corresponding latitude and longitude are 45°10′54.95”N and 9°09′21.09”E, respectively. The spectral means for the selected classes in the ground-truth of the source region are shown in Figure 1.22.

The target region, shown in Figure 1.23, has a spatial size of 350×350 pixels where its upper-left corner corresponds to the coordinates (0, 0) in the original image and its corresponding latitude and longitude are 45°11′23.66”N and 09°08′57.06”E, respectively. The

Table 1.21: Number of pixels per class for Pavia City dataset.

Label	Name	# of available pixels
1	Roads	17494
2	Vegetation	43104
3	Shadows	49703
4	Buildings	18966
Total		129267

spectral means for the selected classes in the ground-truth of the source region are shown in Figure 1.24. Similar to [86], 200 labeled samples per class were selected from the source image for training and only 50% of the target samples were used for test. The number of available pixels per class in the source and target reference data are shown in Table 1.22.

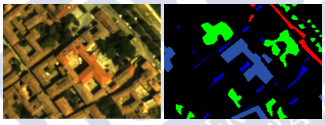


Figure 1.21: False color composite image and reference data for the source domain of the Pavia City dataset.

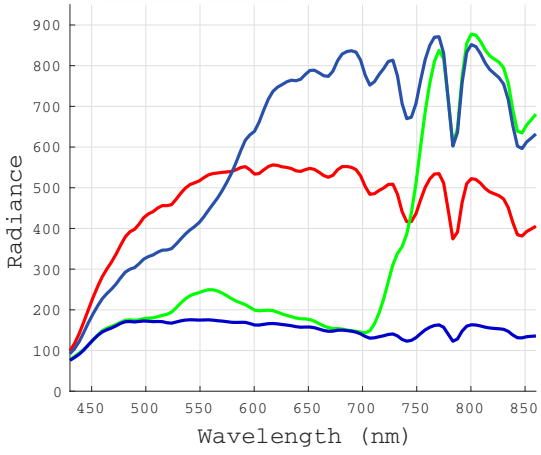


Figure 1.22: Spectral mean values for the selected classes in the Pavia City source dataset. The color code is the same as for Figure 1.20.

Table 1.22: Number of pixels per class for source and target domains of Pavia City dataset.

Name	Source samples	Target samples
Roads	326	2549
Vegetation	1793	6406
Shadows	514	1638
Buildings	1465	17501
Total	4098	28094

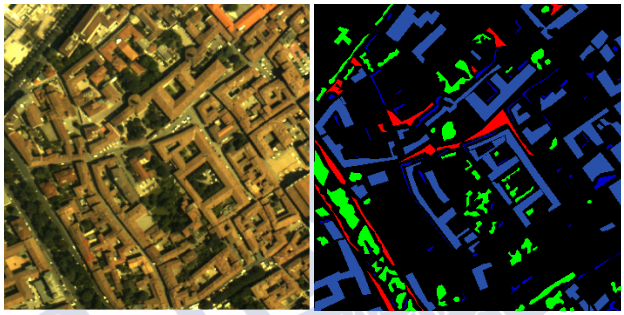


Figure 1.23: False color composite image and reference data for the target domain of the Pavia City dataset.

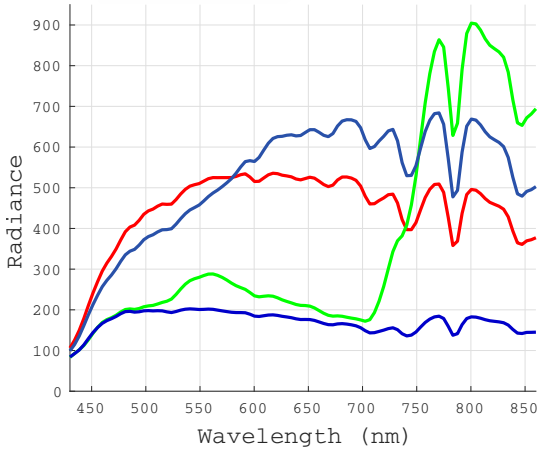


Figure 1.24: Spectral mean values for the selected classes in the Pavia City target dataset. The color code is the same as for Figure 1.20.

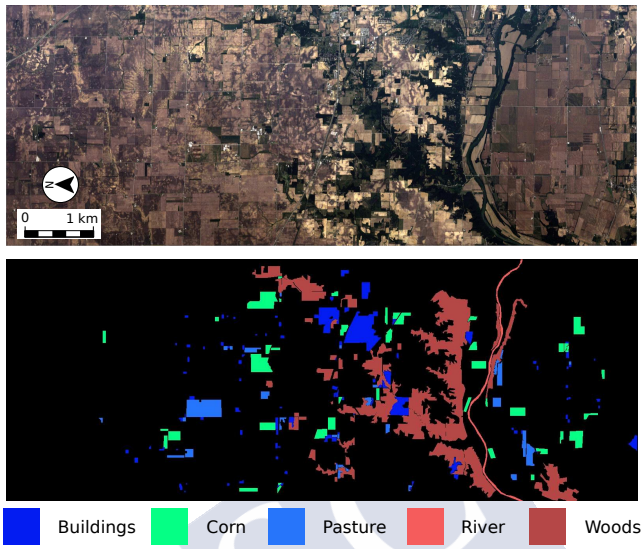


Figure 1.25: False color composite image, reference data and label meaning for the Indiana dataset.

Indiana Dataset

This dataset was acquired by the AVIRIS hyperspectral sensor over the city of Indiana [136] on 12 June 1992. It was collected over the same area as the small Indian Pines image, but spanning a much larger area. It has a spatial dimension of 2678×614 pixels with a spatial resolution of 20 m and covers about 657 km^2. The spectral resolution is of 220 bands ranging 400–2500 nm. The hyperspectral image and its reference data (ground-truth) are shown in Figure 1.25 that was rotated 90 degrees to the left. The total number of classes in the reference data is fifty-eight.

The two disjoint regions used as source and target are shown in Figure 1.26 and Figure 1.27, respectively. On the one hand, the source region has a size of 301×329 pixels where its upper-left corner corresponds to the coordinates (0, 2340) in the Indiana image and its corresponding latitude and longitude are 40°24′02.00”N and 87°00′04.38”O, respectively. On the other hand, the target region has a spatial resolution of 700×614 pixels where its upper-left corner corresponds to the coordinates (0, 1576) in the original image and its corresponding latitude and longitude are 40°31′08.58”N and 86°56′51.82”O, respectively.

Given the two disjoint regions (source and target), 5 out of the 58 available classes were selected for the experiments considering that they should be available in both regions. The number of available pixels per class in the source and target reference data is shown in Table 1.24 and their spectral means are displayed in Figure 1.28 for the source, and in Figure 1.29 for the target. A maximum number of 200 labeled samples per class from the source image were selected for training. For the classes with a number of samples lower than 200, all the available samples were selected for training. Similar to [86], only 50% of the target samples were used for test.

Table 1.23: Number of pixels per class for Indiana dataset.

Label	Name	# of available pixels
1	Buildings	17195
2	Corn	17783
3	Pasture	10386
4	River	3110
5	Woods	63562
Total		112036

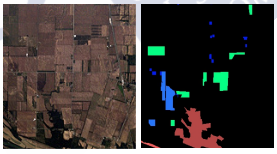


Figure 1.26: False color composite image and reference data for the source domain of Indiana dataset.

Table 1.24: Number of pixels per class for source and target domains of Indiana dataset.

Name	Source samples	Target samples
Buildings	432	15003
Corn	3062	11116
Pasture	1742	4154
River	197	540
Woods	4543	54230
Total	9976	85043

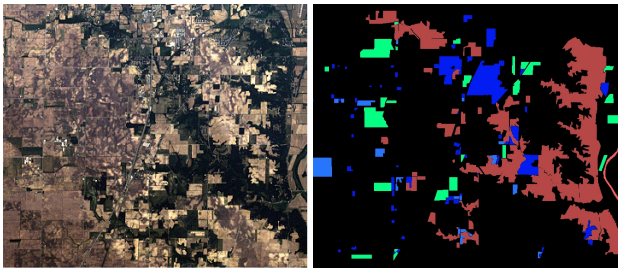


Figure 1.27: False color composite image and reference data for the target domain of the Indiana dataset.

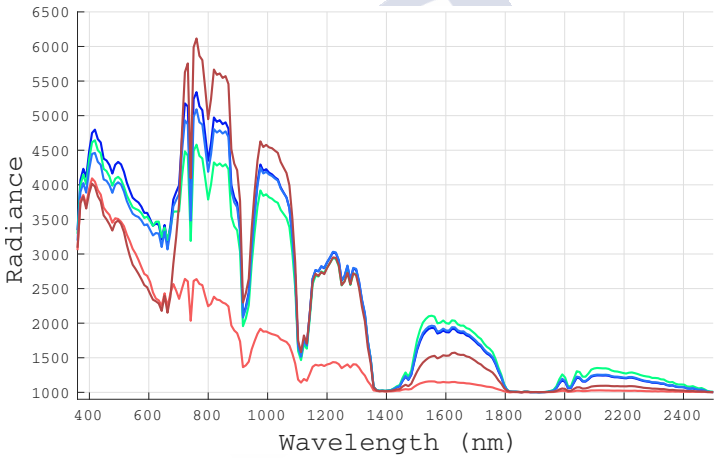


Figure 1.28: Spectral mean values for the selected classes in the Indiana source dataset. The color code is the same as for Figure 1.25.

1.4 Work summary

Technological advances are continually being added to remote sensing imaging devices as spaceborne and airborne platforms. This has led to a considerable increase on the demand of geoinformation by both governments and the private sector [137]. In the same way, the proliferation of small unmanned airborne platforms, such as UAVs, has also contributed to the wide availability of this kind of information [138]. As a result, the number of new datasets to be analysed and their size are continuously growing.

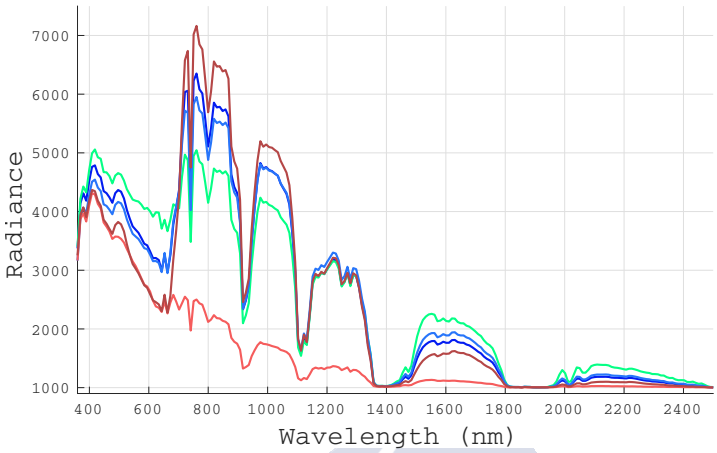


Figure 1.29: Spectral mean values for the selected classes in the Indiana target dataset. The color code is the same as for Figure 1.25.

Nowadays, the use of UAVs for capturing remote sensing images has become very popular. Due to the short distance between the sensor and the area to be captured, UAVs offer high spatial resolution compared to other platforms such as satellites or high flying planes. This feature enables very dense multispectral and hyperspectral imaging. In addition, this type of device has allowed even small companies to obtain their own datasets, thanks to their low cost. As a consequence, many areas, including the analysis of vegetation, have become increasingly interesting for researchers. In this thesis, remote sensing images of the Earth's surface covering buildings but also vegetation zones and crops are analyzed.

The efficient classification of multi and hyperspectral images in the field of remote sensing is an open problem. The use of ML and, in particular, DL-based architectures, such as DNNs or CNNs, has allowed the design of new solutions during the last few years [71]. In this thesis, the design of efficient remote sensing image classification schemes based on ML and DL architectures has been carried out. Spectral, spatial and textural information is extracted in order to adapt them to the specific remote sensing field. As the resulting techniques entail a fairly high computational cost, different modifications have been applied to decrease the execution time.

In this section, the different contributions of this thesis will be presented. We will start with a scheme proposed for the classification of hyperspectral images based on ELM, a SLFN with fixed weights that reduces the training time. Next, a scheme that combines spectral,

spatial, and texture information together with a segmentation algorithm to identify different vegetation species and non-vegetation structures with a final step of SVM-based classification is addressed. Next, the discussion focuses on different implementations of a CNN-based scheme for the classification of hyperespectral images. Then, the DA problem is studied and TCANet, a scheme combining CNNs with an algorithm specifically designed for DA called TCA is presented. The CNN filters are calculated using TCA thus reducing the computational cost. Finally, HypeRvieW, a tool developed to analyse and process multi and hyperspectral images, is also presented as it is part of the contributions of this thesis.

The field of ANN is continuously growing and it has been successfully used for classification in remote sensing [9, 10]. ANN are not focused on statistical relationships [139]. Nevertheless, their high computational cost drives the process of finding new less demanding schemes. Multilayer feedforward neural networks with supervised learning use an iterative process to be trained. This traditional iterative process and the number of neurons and layers that compose the network influence the learning speed of the feedforward neural networks. These features along with other issues such as slow convergence can be seen as a bottleneck. The reason for this slow convergence is the use of gradient descent algorithms as the basis for tuning all the learning parameters of the neural network [51]. To avoid these problems, Neural Networks with Random Weights (NNRW) were proposed.

Randomness is a way of reducing the training time of the networks. It has been introduced into neural networks since the era of the perceptron model [45, 14]. Recently, Huang et al. [15] defined a class of SLFN, called ELM, that has gained importance because it uses a non-iterative process to train the network, thus reducing the computational cost. Based on the minimum norm least-squares solution of a general linear system and the Moore-Penrose generalized inverse [140], Huang et al. propose a solution with interesting features. Different from other learning algorithms that cannot reach the global minimum because of local minimums, the solution proposed by Huang et al. allows to reach the smallest training error. Furthermore, the solution obtains the smallest weights. Based on the Bartlett's theory on the generalization of performance of feedforward neural networks [141], which shows that to have a good generalization performance on the network we have to maintain the norm of the weights small, the solution proposed by Huang tends to have better generalization properties than a traditional gradient-based learning algorithm.

Huang et al. [16] also proposed the use of a kernel function instead of random weights in the hidden layer. That change leads to an ELM version, named KELM, with better generaliza-

tion capabilities compared to the use of random weights proposed in the first version of ELM. With this new version, the variation of accuracy produced by the randomness in different trials is avoided.

In addition to KELM, a wide variety of kernel-based methods [142] exists, but one of the most extended is Support Vector Machine (SVM) [143]. Whereas SVM uses a kernel function to map the input data to a some higher order feature space to find the best hyperplanes that separate the classes, KELM uses the kernel function to obtain the feature mapping (the weights of the hidden layer) and then computes the output weights through the use of the Moore-Penrose generalized inverse to solve a simple liner system. Both methods, SVM and KELM, introduce parameters that need to be tuned. Several papers [144, 145, 146] compare the features of both methods SVM and ELM and almost all conclude that, although both obtain very close accuracy values, ELM and its variant KELM present less computational complexity which helps them to be faster.

In this thesis, a scheme named KELM-EMP that combines spectral and spatial information which is later used by a KELM-based classifier is proposed, as it is presented in chapter 2. This scheme is divided into 4 main steps: dimensionality reduction, spatial information extraction, spectral-spatial information merging and, finally, the classification step.

The first step of the scheme, the dimensionality reduction, is required as the vast amount of information provided by hyperspectral images can increase the processing complexity and the execution time. The reduction of the dataset dimensionality can help decrease them by removing redundant information. In the proposed scheme this step is carried out by a PCA algorithm [19].

Mathematically, several method can be used to compute the PCA. In this thesis, an approach that uses Eigenvalue Decomposition (EVD) of the covariance matrix of the dataset was selected. Furthermore, a comparison between this approach and an iterative PCA method based on the Gram-Schmidt (GS) re-orthogonalization process called GS-PCA [147] was also performed. The big difference in execution time caused the iterative version to be discarded.

The second step is the spatial information extraction. MPs are widely used in remote sensing for feature extraction in image classification [148]. They extract spatial information by sequentially applying morphological transformations with a Structuring Element (SE) of increasing size over the bands of the image. In this scheme, morphological opening and closing operations are applied over each Principal Component (PC) obtained by the PCA algorithm for extracting spatial information. The number of MPs obtained for each PC will

be one (corresponding to the original data) plus twice the number of different sizes for the SEs. As a result, an EMP containing the spatial information of the image is generated from the concatenation of all the MPs [23].

The third step of the proposed KELM-EMP scheme is the merging of the spatial and the spectral information. In particular, a weighted concatenation of both the EMP and the spectral information is performed. Then, in the last step, a selected number of samples are used to train the KELM-based classifier, while the other pixels are used to obtain the final classification. To avoid the appearance of isolated classified pixels, a regularization process, using eight neighbours, is performed after the classification.

The full scheme, called KELM-EMP-S, and a derived scheme replacing KELM by ELM, named ELM-EMP-S, were compared in terms of accuracy against several schemes that use ELM and SVM as classifiers and watershed or EMP to extract spatial information. In addition, a comparision between KELM-EMP-S and ELM-EMP-S in terms of execution time was performed. Three different implementations were developed for the execution time evaluation. The first one is a sequential implementation and it was executed in CPU. For the second implementation, several Open Multi-Processing (OMP) directives were applied to execute the code using as many threads as were available in the multicore CPU. The last implementation was developed using CUDA and executed in GPU.

Regarding the precision comparison, the best result was obtained using seven PCs as input to the EMP algorithm. The KELM-based scheme proposed, outperforms all the other ones achieving accuracies up to 99.83% for the Pavia University dataset. In terms of execution times, the KELM-EMP-S scheme in GPU version obtained a speedup of over 17.04$\times$ compared with the CPU version.

In this thesis, a particular interest is placed on the classification of vegetated areas. The spectral response of these areas can be interpreted using simple methods such as the calculation of vegetation indices [149], e.g. Normalized Difference Vegetation Index (NDVI) [150]. Nevertheless, these indices are not suitable to classify vegetation. On the other hand, it is well-know that in the remote sensing field, combining information coming from the spatial structures present in the input image and spectral features available for each pixel of the image improve the classification results, as explained above [151].

For remote sensing image analysis, in the case of vegetation species identification, the use of texture information can help in the classification process. This kind of information allows characterizing the areas of interest. Textures are combined in many cases with other features

to improve the classification accuracy [152, 153]. Vegetation indices and morphological measures are examples of these features. For example, for the detection of the extent of trees and shrubs, the Canopy Height Model (CHM) is the one most commonly used. Surface reconstruction by image matching can also be used to estimate CHM. It is achieved by exploiting the redundancy of multiple overlapping aerial images [154, 155]. CHM is not used in this paper as for the available datasets the required olverlapping images are not available.

Unsupervised algorithms such as k-means can be used for texture feature extraction. This clusterization algorithm, separate pixels into clusters. K-means uses the Euclidean distance, as separation criteria, in an iterative process to minimize the distance between the centers of the clusters (called centroids) and the other pixels of the image. The iterative process is performed until a certain criterion (a fixed number of iterations or an error value) is reached. As a result, a texture dictionary is built. Then, this dictionary is used by feature encoding algorithms like Fisher Vector (FV) [28] or Vector of Locally Aggregated Descriptors (VLAD) [31] to characterize the texture of a certain region of the image.

For the classification of a image using textures it is necessary to delimit regions over which the texture features are computed. Most of the vegetation classification methods proposed in the literature use regular patches [60, 156, 152]. In other cases, segmentation or object detection algorithms for dividing the image into regions are used [157, 158]. A technique commonly used for the extraction of uniform regions in images is the segmentation based on superpixels [32, 33]. A superpixel is a set of neighboring pixels (segment) which are similar in terms of low-level properties (such as spatial proximity, color, intensity, or other criteria). Superpixel-based segmentation methods allow to obtain superpixels with the same size and regularity throughout the image. This provides a convenient and compact representation of the images that allow to reduce the computational cost of the processing algorithms if they are applied at the segment level instead of at pixel level [159]. In the scheme presented in this paper, a texture feature vector is computed for each superpixel.

The combination of spectral-spatial features and textural features represents an increase in computational cost for the classification process since it has to work with more information per pixel. To reduce this computational cost, a classification at the level of superpixel instead of at the level of pixel is carried out, thus reducing the computational cost.

In chapter 3, a spectral-spatial texture-based classification scheme to solve the problem of identifying several vegetation species and non-vegetation structures on multispectral images

was proposed. It is applied over Simple Linear Iterative Clustering (SLIC) segments. The algorithm is divided into two main steps: feature extraction and hierarchical classification.

In the first step, spectral-spatial and texture features are extracted from the images. Initially, an EMP containing the spectral-spatial information is generated through mathematical morphological opening and closing operations. These operations use circular structuring elements of different sizes to build the MPs of each band of the image. The EMP is the set of resulting MPs. Regarding the texture information, the *k*-means clusterisation algorithm was used. This unsupervised classifier separates pixels into a number of classes decided by the user.

The last step for the feature extraction is to combine both the spectral-spatial and the texture features with the superpixels obtained by the SLIC algorithm. On the one hand the segments are combined with the spectral-spatial information. So, for each segment and each band of the image calculated from the EMP, the average and standard deviation values for the pixels in the segment are computed. On the other hand the texture data is combined also with the segments to extract the variation of individual pixels inside segments to distinguish among vegetation canopy. To do this, based on the information extracted by *k*-means, VLAD generates a vector representation for each superpixel.

For the classification step, 4 different SVM classifiers were used in a hierarchy. It allows the first classifiers to separate the data into large groups with well-differentiated features, while the last classifiers are responsible for a fine classification. The first SVM is trained using spectral-spatial features to discriminate between vegetation and non-vegetation. Then, all the superpixels classified as non-vegetation go to the second SVM, that divides superpixels in water or non-water using spectral-spatial features. Next, those pixels classified as non-water go to the third SVM. It is trained with the spectral-spatial features and discriminates between the different classes of non-vegetation structures. Finally, all the superpixels previously classified as vegetation go to last SVM that is in charge of the discrimination between the different classes of vegetation. This SVM is trained using the texture features.

For the experiments, different multispectral datasets were used. They were captured in different geographical areas of the Oitavén and Xesta River basins by the MicaSense RedEdge multispectral camera. A comparison in terms of accuracy between the proposed scheme and a pixel-wise SVM classifier was performed. As a result, our proposal obtains an increase in OA of 15%. The accuracies of the first two SVMs, that identify vegetation and water respectively, allow the hierarchical scheme to achieve higher global precision than a single SVM scheme.

Autoencoders (AEs) [160] are a type of DL-based methods for classification applied to remote sensing. They play an important role in unsupervised learning and transfer learning schemes [161, 162], as they were specifically designed to perform FE and dimensionality reduction. An AE aims to learn, in an unsupervised way, a coded representation of the input data. It can be seen as a type of neural network made up of three different layers: an input layer and an output layer of s neurons each, and a hidden layer of t neurons between them, where $s > t$. It can be used as a main building block for generating more complex models such as: Stacked Autoencoders (SAEs) [163, 75], where several AE are concatenated; DAEs [82, 164], where a certain level of noise is added to the input data in the training process to obtain a more robust network; Variational Autoencoders (VAEs) [165], that are learning models that mix neural networks with probability distributions to build generative models that are capable of producing synthetic data that follow the same patterns as the datasets which they were feed; etc.

Another deep networks model is DBN [77, 166, 167]. Its main building blocks are layers of Restricted Boltzmann Machine (RBM) [168] that comprise a single visible layer and a hidden layer. As in the case of AEs, training is performed layer by layer in a separate way and the main objective is to learn progressively higher-level features from original data in an unsupervised manner. Once the network has been pre-trained, the learned weights are used as the initial weights in a traditional backpropagation driven feed-forward network training. This initialization of the weights helps the traditional backpropagation to reach a better regions of parameters search space.

Generative Adversarial Networks (GANs) [169, 170] are another deep network model and together with DBN, they are an approach to generative modeling. It means that the model can automatically learn the patterns in the input data so that it is able to generate new examples which could have been drawn from the original dataset. There are two submodels inside a GAN: a generative model for generating new examples and a discriminative model for classifying whether generated examples are real, from the original dataset or fake, generated by the generative model. Initially, the discriminative model is only capable of estimating whether the input sample is real or fake, but some changes have been proposed to do it suitable to multi-class classification [170, 171].

The training phase for the previous deep networks models is performed in an unsupervised way. However, other deep networks models such as CNNs [172, 173] need reference information to be trained in a supervised manner.

During the learning process in a traditional CNN, each of their convolution layers extracts the contextual two-dimensional spatial features of the input data. First, a two-dimensional patch or window going through the entire image is convoluted using convolutional filters, resulting in a new feature map. Then a pooling operation is applied to the featured map coming from the convolutional layer to reduce the redundant information. This operation can also reduce its size and therefore the number of parameters of the network. The two previous operations can be repeated in a loop, being the number of iterations the deep of the network, considering that the deeper the network, the more complex the extracted features will be. Finally, a set of fully connected layers are feed with the flattened output of the last pooling layer to perform the final classification.

Like many of the networks mentioned in this thesis, the training process is performed using backpropagation. Starting on the fully connected layer and ending on the first convolution of the network, the updated weights are propagated through the network. So, the learning process and the backpropagation steps are repeated until the network achieve a reasonable level of learning. In this training phase, we can use all the patches at the same time or group them into smaller sets, called batches, so that the training is not so complex.

The development of frameworks that allow combining the strong capacity of deep network models and the computational power of GPUs helped to build new powerful schemes. TensorFlow [90], PyTorch [91], Caffe [39] and Keras [93] are some examples of currently active frameworks and some of the most used. The vast majority of the frameworks available today use a Python API though the most widely used ones often have APIs for other programming languages as C, C++ or Java.

In chapter 4, a GPU implementation of a CNN-based spatial-spectral scheme was developed for the supervised classification of remote sensing datasets using two deep learning libraries, Caffe (now included into PyTorch) and cuDNN. A comparison against a CPU version of the same scheme is also performed. Four well know datasets are considered for the experiments: Pavia University, Indian Pines, Salinas, and Pavia Centre. The configuration parameters were determined by performing experiments varying the number of PCs, the batch size and the filter size for the code executed in CPU, and also for both GPU implementations. The main parts of the scheme are: dimensionality reduction, patch extraction, feature learning and classification.

Initially, to reduce the high computational cost of working with datasets with hundreds of bands of information and thus be closer to a real time response, a PCA dimensionality

reduction algorithm was applied. Different experiments led to the selection of four PCs as the best configuration as output of the PCA algorithm.

The next step is to extract a two-dimensional patch around each of the training pixels of the dataset. Each one will be considered a sample and it will be used as the unit of information during the convolution and classification phases. Next, the feature learning is carried out by a CNN. In this scheme, it comprises a convolutional layer followed by a pooling layer. The last step is to obtain the final classification and it is performed by a fully connected neural network that is located immediately after the pooling layer. This network is a typical multilayer perceptron with a hidden layer and an output layer.

For the GPU implementation, the last two steps of the scheme (feature learning and classification) were carried out using two different libraries: Caffe and cuDNN. A comparison, in terms of speed and accuracy between the CPU version and the two GPU implementations, was performed. Regarding the GPU implementations, the block size is an important parameter as it defines the number of threads working at the same time. So, a comparative study of the time based on the block size was carried out. The use of the cuDNN library greatly reduced the computational time while increasing precision. Speedups of up to 74.22$\times$ were achieved using the cuDNN library compared to a CPU version of the general scheme. Regarding the accuracy, the GPU implementations of the scheme obtain similar results as the CPU version. A similar scheme but with two convolution and pooling layers instead one were executed using the cuDNN library. This new scheme improved the accuracies obtained by previous GPU implementations with a slight increase in computation time.

The lack of high quality reference data [174, 175] poses a particular challenge in the classification of the images. The problem is more complex if images belonging to different spatial areas need to be classified or if they were taken by different sensors or at different times. In all these cases, the spectral shift between the different images, produced during in-flight data acquisitions due, for example, to instrumental pressure, temperature or vibrations, could reduce the accuracy of a joint classifier [175, 176]. Furthermore, because the scarcity of available reference information affects the success of the classification task, and since the manual labelling process is very expensive [177], the need to find techniques that take advantage of all the available reference information gains particular relevance.

Considering that two images are involved, the different approaches for DA can be grouped as [73]:

1. Selection of invariant features. From the original set of features, those that are invariant between the domains are used to train the classifier. Thus, assuming that the features that are affected by shift are removed, the new feature space gives a better consistence between the domains [178, 179, 180, 37].

2. Adaptation of the classifier. This family of techniques are based on semisupervised learning. It uses a subset of unlabeled samples from the target domain to adapt a classifier model trained with samples from the source domain. During the process, the data distributions remain unaltered [181, 182].

3. Adaptation of the classifier by active learning. As in the previous approaches, a semisupervised approach is taken, but in this case a reduced number of labeled samples of the target domain are provided by the user. These samples have to be carefully selected to lead the classifier model to perform well on unlabeled target samples [183, 184, 185].

4. Adaptation of the data distribution (also know as representation learning or FE). In this case the different techniques aim to reduce the shift between the data distributions of both source and target domains before the classification. Within these techniques we can find some as simple as CORAL [37], based on the alignment of the covariance of both domains, or other more complex based on neural networks such as Stacked Denoising Autoencoderss (SDAEs) [36].

In this thesis a new scheme for DA based on the last approach is proposed. The aim is to find the mapping function that best describes the data belonging to the two domains involved. The approach is based on CNNs. The resulting network uses an iterative learning process that is time consuming process. The cost, in particular the training time, could be reduced by using methods such as TCA to learn the coefficients of the convolutional filters on each layer of the CNN, since the back propagation step is avoided.

In chapter 5, the scheme called TCANet is proposed. The scheme classifies classifies an image (target) using information from a different one (source) and includes two CNNs where the convolutional filters were replaced by fixed filters computed using TCA. Since TCA is sensitive to normalization, a conditional correlation alignment using the second-order statistic is applied at the beginning of the scheme to reduce the domain shift between the source and the target. The final classification process was performed by a SVM, trained using the adapted source data to obtain the target reference map.

Since TCANet simulates the behaviour of a CNN, the first step is to prepare the input data according to that type of network. So, a set of patches from both images (source and target) are extracted. Each patch corresponds to a three-dimensional spatial window $(D \times D \times B)$ around a selected pixel, being D the spatial width and height of the window, and B the number of bands of the corresponding image. Then, a reshape process to obtain a final window of two-dimensions $(D^2 \times B)$ is performed.

Once the input for first convolution is ready, the next step computes the filters. So, all the reshaped patches are factorized into non-overlapping blocks and used as input of the TCA algorithm to perform the computation of the filter coefficients. Due to the DA characteristics of TCA, the convolutional filters are now ready to perform domain adaptation. Then, all the two-dimensional windows (reshaped patches) are factorized into overlapping blocks and processed, one-by-one into the CNN, to compute the output of the first convolution. All the previous steps are repeated for the second convolution using the output of the first layer of the CNN as input to the next one. As a final step, a feature extraction process based on heaviside functions is performed after the second CNN. Finally, a SVM is trained with the transformed features of the selected training samples from the source image to estimate the reference map of the target image.

The proposed scheme was compared to several methods using different datasets. In all the cases, two disjoint regions were selected as source and target. The results show that TCANet outperforms other methods presented in the literature, especially those that only use information coming from the source domain for the training process. For methods that use information from both domains, only those based on DAE obtain slightly better accuracy but with a worse standard deviation. Compared to others methods, TCANet does not require neither regularized parameters nor numerical optimization solvers due to the use of fixed filters computed with the information extracted from both the source and the target domain.

The use of specialized tools for the analysis and processing of multi and hyperspectral datasets is a requirement for most standard users. In the case of free software and open access tools, Geographic Resources Analysis Support System Geographic Information System (GRASS GIS) [186] is one of the most complete Geographic Information System (GIS) software. It has been under continuous development since 1982. The latest stable release version (LTS) is GRASS GIS 7, which is available since 2015 and that is currently available for download in version 7.8.5 (December 21, 2020). It is built for vector and raster geospatial data management, geoprocessing, spatial modeling and visualization. GRASS GIS is multi-platform,

it has been developed for the three main operating system: Linux, Mac and Windows. It has over 500 modules to process the information both graphically or via command line. Among the different modules are image analysis and processing algorithms such as both supervised and unsupervised classification.

Another one of the most commonly used free software and open access tools is QGIS [187] that started life back in 2002 and that is currently in its version 3.16. This software is available for a wide variety of platforms, including Android. In the case of the Android platform, it is a reduced version of the software that would facilitate field work with functions such as digitization or attribute search, among others. QGIS allows the reading of a large number of image formats, both raster and vector, and provides powerful tools for data analysis through both its own tools and through integration with the GRASS GIS application.

The two tools mentioned above are the most popular in terms of free software and open access, but there are many others such as: gvSIG [188], that began in 2004 as large project in terms of financial and development resources founded by the Regional Council for Infrastructures and Transportation of Valencia (Spain), although since 2010 it has been managed by the gvSIG association; or GeoDa [189], that since its inception in 2003 has exponentially increased the number of users to over 360.000.

In terms of proprietary software, there are three great tools: ENVI [190], eCognition [191] and ArcGIS [192]. Any of them offers access to a wide catalogue of algorithms that facilitate both the analysis and the processing of remote sensing images and, like some of the free software and open access tools, allow to extend its functionality through the Python programming language.

In chapter 6, an open source desktop application for remote sensing dataset analisys and processing, called HypeRvieW [193], is described. This software arose from the need to find some place to group all the variety of algorithms developed inside our research group and at the same time be able to combine them into new schemes. Another reason is that all the algorithms developed in our research group were programmed in C or C++ language and none of the available tools (with a commercial or free license) allowed the integration of external software in that language, or at least not in an easy way. As a result, HypeRvieW incorporates a set of tools that allow remote sensing dataset processing and analysis.

Regarding the analysis tools, HypeRvieW allows a deep analysis of the input dataset by visualizing each of the bands in a separate way. In addition, HypeRvieW also has a tool that allows visualizing the spectral signature of a pixel selected by the user and optionally

comparing it to another pixel in the dataset or to one of the spectral signatures included in the tool. These spectral signatures corresponds to a wide variety of materials and vegetation such as: brick, window glass, white marble, pine, eucalyptus, grass, etc.

One of the most interesting tools within HypeRvieW is its reference information generator. This tool allows to create or edit the reference data of a dataset. The user can work directly on a full dataset or use a segmentation algorithm to work with the different segments. The tool includes several drawing tools such as polygons or brush strokes in the style of a drawing program. So, the user can classify pixel by pixel (or segment by segment) or select a group of pixels (or segments) using a polygon and classify all of them with the same class. Every time a pixel or a segment is painted, it is assigned to a certain, previously created, class. The user can create as many classes as needed, assigning them a name and a color.

With respect to the processing tools, HypeRvieW allows users, in a very simple way, to include their own algorithms, programmed in C or C++ language, to the set of algorithms already available in the tool. Among the algorithms included by default we can mention: ELM, EMP, PCA, SVM, etc. The algorithms can be combined to perform a more complex analysis.

HypeRvieW has a graphical interface tool that enables the user to create supervised classification schemes. The design of these new schemes is carried out by joining the different algorithms, identified in the tool as boxes with inputs and outputs. The number of inputs and outputs depends on the algorithm. Thus, if an algorithm needs two input images and produces a single output, it will be represented by a box with two inputs and one output. The input images are also symbolized as boxes, which consist of a single output. There is also a box to shown the classification results both graphically and in text format. In the graphic format, the complete image will be displayed, colouring each of its pixels with the color corresponding to the class that has been assigned to it. For the text format, the accuracy results for the classification, the statistics regarding the number of pixels per class, and all those messages that the different algorithms involved in the experiment have generated will be displayed. For each of the boxes that symbolize an algorithm, the parameters corresponding to that algorithm can be specified. The concatenation of any two algorithms A and B is performed by joining the output of the box that symbolizes algorithm A with the input of the box that symbolizes algorithm B. This modularity gives the tool great flexibility when developing new schemes.

In addition to the features discussed above, HypeRvieW allows other basic operations such as: zooming in and out over the bands of the datasets, saving data in several formats,

working with two datasets at the same time, cropping on the image, rotating and scale the image, save the results of a classification, compute several biophysical indexes over an image, generate a false color image based on the characteristics of the sensor used to capture the datasets, etc.

This tool has become very relevant for the research groups and it is continuously evolving as it facilitates the comparison of different implemented algorithms such as classification schemes in a more agile way. It also allows the researchers to access the different algorithms developed within the group without the need to adapt them to their own codes, and makes the introduction of new students to the research tasks carried out by the group easier. From the transference point of view the tool was essential to explain our work to different companies, such as Babcock, that supported part of this thesis.

1.5 List of publications

1.5.1 International Journals

- [101] A. S. Garea, Á. Ordóñez, D. B. Heras, and F. Argüello, "HypeRvieW: an open source desktop application for hyperspectral remote-sensing data processing," *International Journal of Remote Sensing*, vol. 37, no. 23, pp. 5533–5550, 2016
  Impact factor (JCR 2016): 1.724. Q2.
  Category: Imaging Science & Photographic Technology. Rank: 13/26.
  Impact factor (SJR 2016): 0.804. Q1.
  Category: Earth and Planetary Sciences. Rank: 338/1646.

- [95] A. S. Garea, D. B. Heras, and F. Argüello, "GPU classification of remote-sensing images using kernel ELM and extended morphological profiles," *International Journal of Remote Sensing*, vol. 37, no. 24, pp. 5918–5935, 2016
  Impact factor (JCR 2016): 1.724. Q2.
  Category: Imaging Science & Photographic Technology. Rank: 13/26.
  Impact factor (SJR 2016): 0.804. Q1.
  Category: Earth and Planetary Sciences. Rank: 338/1646.

- [98] A. S. Garea, D. B. Heras, and F. Argüello, "Caffe CNN-based classification of hyperspectral images on GPU," *The Journal of Supercomputing*, vol. 75, no. 3, pp. 1065–1077, 2019

Impact factor (JCR 2019): 2.469. Q2.
Category: Computer Science, Theory & Methods. Rank: 31/108.
Impact factor (SJR 2019): 0.432. Q2.
Category: Computer Science. Rank: 973/4116

- [100] A. S. Garea, D. B. Heras, and F. Argüello, "TCANet for domain adaptation of hyperspectral images," *Remote Sensing*, vol. 11, no. 19, p. 2289, 2019
  Impact factor (JCR 2019): 4.509. Q2.
  Category: Remote Sensing. Rank: 9/30.
  Impact factor (SJR 2019): 1.422. Q1.
  Category: Earth and Planetary Sciences. Rank: 140/1377.

1.5.2 International Conferences

- [102] A. S. Garea, D. B. Heras, and F. Arguëllo, "An open source desktop application for classification of remote sensing data," in *2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 1. IEEE, 2015, pp. 316–321

- [99] A. S. Garea, D. B. Heras, and F. Argüello, "GPU classification for hyperspectral images based on convolutional neural networks," in *Computational and Mathematical Methods in Science and Engineering*, J. Vigo-Aguiar, Ed., 2017, pp. 912–923

- [4] J. López-Fandiño, A. S. Garea, D. B. Heras, and F. Argüello, "Stacked autoencoders for multiclass change detection in hyperspectral images," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 1906–1909

- [97] P. G. Bascoy, A. S. Garea, D. B. Heras, F. Argüello, and A. Ordóñez, "Texture-based analysis of hydrographical basins with multispectral imagery," in *Remote Sensing for Agriculture, Ecosystems, and Hydrology XXI*, C. M. U. Neale and A. Maltese, Eds., vol. 11149, International Society for Optics and Photonics. SPIE, 2019, pp. 225 – 234

1.5.3 National Conferences

- [96] A. S. Garea, D. B. Heras, and F. Argüello, "Clasificación de imágenes de teledetección mediante ELM kernel y perfiles morfológicos en GPU," in *Jornadas SARTECO 2016*. Ediciones Universidad de Salamanca, 2016, pp. 81–89

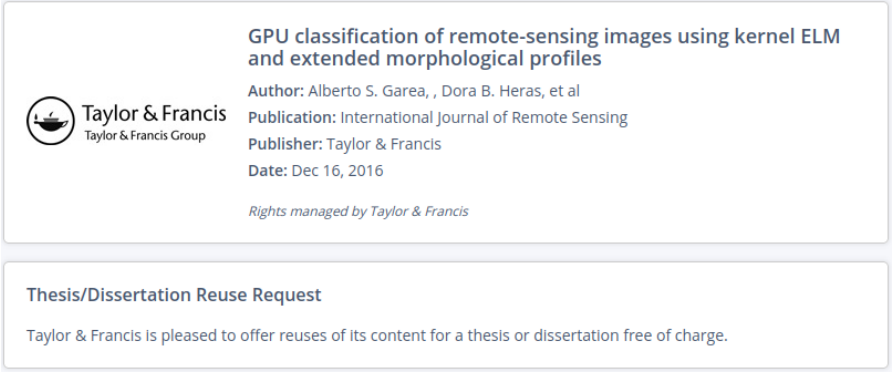
USC
UNIVERSIDADE
DE SANTIAGO
DE COMPOSTELA

Chapter 2

GPU Classification of Remote Sensing Images using Kernel ELM and Extended Morphological Profiles

A. S. Garea, D. B. Heras, and F. Argüello, "GPU classification of remote-sensing images using kernel ELM and extended morphological profiles," *International Journal of Remote Sensing*, vol. 37, no. 24, pp. 5918–5935, 2016

https://doi.org/10.1080/01431161.2016.1251629



USC
UNIVERSIDADE
DE SANTIAGO
DE COMPOSTELA

Chapter 3

Texture-based Analysis of Hydrographical Basins with Multispectral Imagery

P. G. Bascoy, A. S. Garea, D. B. Heras, F. Argüello, and A. Ordóñez, "Texture-based analysis of hydrographical basins with multispectral imagery," in *Remote Sensing for Agriculture, Ecosystems, and Hydrology XXI*, C. M. U. Neale and A. Maltese, Eds., vol. 11149, International Society for Optics and Photonics. SPIE, 2019, pp. 225 – 234

https://doi.org/10.1117/12.2532760

USC
UNIVERSIDADE
DE SANTIAGO
DE COMPOSTELA

Chapter 4

Caffe CNN-Based Classification of Hyperspectral Images on GPU

A. S. Garea, D. B. Heras, and F. Argüello, "Caffe CNN-based classification of hyperspectral images on GPU," *The Journal of Supercomputing*, vol. 75, no. 3, pp. 1065–1077, 2019

https://doi.org/10.1007/s11227-018-2300-2

USC
UNIVERSIDADE
DE SANTIAGO
DE COMPOSTELA

CHAPTER 5

TCANET FOR DOMAIN ADAPTATION OF HYPERSPECTRAL IMAGES

A. S. Garea, D. B. Heras, and F. Argüello, "TCANet for domain adaptation of hyperspectral images," *Remote Sensing*, vol. 11, no. 19, p. 2289, 2019

https://doi.org/10.3390/rs11192289

USC
UNIVERSIDADE
DE SANTIAGO
DE COMPOSTELA

Chapter 6

HypeRvieW: an Open Source Desktop Application for Hyperspectral Remote Sensing Data Processing

A. S. Garea, Á. Ordóñez, D. B. Heras, and F. Argüello, "HypeRvieW: an open source desktop application for hyperspectral remote-sensing data processing," *International Journal of Remote Sensing*, vol. 37, no. 23, pp. 5533–5550, 2016

https://doi.org/10.1080/01431161.2016.1244363

USC
UNIVERSIDADE
DE SANTIAGO
DE COMPOSTELA

Conclusions

The efficient spectral-spatial classification and domain adaptation of n-dimensional images using neural networks and deep learning were addressed in this thesis. The focus was on designing and developing new schemes by producing good classification results in terms of accuracy and efficient computation in commodity hardware. In the following, we summarize the main contributions of this thesis and explain how the objectives have been met:

1. *Efficient spectral-spatial classification schemes based on ELM, a SLFN, have been proposed, as well, as the corresponding CUDA implementations for GPU.*

   The first proposed scheme, named ELM-EMP, performs dimensionality reduction of the hyperspectral image by PCA and then, the spatial information for each band is extracted using morphological operations of opening and closing by reconstruction, thus creating an EMP. Later, a weighted concatenation combines both the spatial and the spectral information. The final classification is carried out by an EMP algorithm.

   The second scheme, named KELM-EMP, performs the final classification with a version of ELM based on kernels (KELM). It uses a kernel function in the hidden layer of the ELM algorithm instead of random weights, thus avoiding the wide variation in classification accuracy in different trials produced by the random weights. Moreover, a spatial regularization process is applied to the previous schemes (ELM-EMP-S and KELM-EMP-S) outperforming them.

   ELM is an extremely suitable algorithm to be implemented on GPUs and other parallel architectures as it is based on matrix operations. Efficient CUDA implementations for the proposed classification schemes have been obtained. The use of different CUDA libraries, as well as the implementation of our own kernel functions, contribute to reducing the execution time up to 17 times compared to the CPU version for the Pavia

University image. Regarding classification results, the accuracy obtained by the proposed schemes ELM-EMP-S and KELM-EMP-S reaches 99.83% for the the same image, outperforming the state-of-the-art spectral-spatial classification schemes based on ELM.

2. *A supervised texture-based classification scheme for the analysis of hydrographical basins with multispectral images has been proposed.*

   The scheme was proposed to analyze the presence of different vegetation species and artificial structures in the riversides by using 5-band multispectral images obtained by an UAV. The classification scheme extracts spatial-spectral information by means of a segmentation algorithm based on superpixels called SLIC and by computing morphological operations over the bands of the image in order to generate an EMP. The texture features are extracted by computing k-means and VLAD. The classification is performed by a hierarchical structure made up of 4 SVM classifiers and is featured not over pixels but over superpixels that represent the segments obtained by the SLIC algorithm, thus reducing the computational time.

   The experimental results over four multiespectral datasets from Galician riversides show that the proposed scheme improves other classification methods achieving very high accuracy results. In particular, accuracies of up to 94.61% were obtained in comparison to values of up to 82.04% in the case of a standard SVM classifier.

3. *A DA technique based on TCA to classify hyperspectral datasets has been proposed.*

   The scheme simulates the behaviour of a conventional convolutional neural network-based scheme, except that instead of applying the widely used backpropagation technique for the learning process, it is replaced by a TCA technique. This way the computational cost of the DA technique is lower than a CNN-based DA scheme.

   The TCANet scheme is divided into four main steps: conditional correlation alignment, patch extraction, feature extraction, and classification. The first step performs a conditional normalization of the source and the target datasets based on the covariance alignment of both source and target distributions, minimizing the domain shift. The second step extracts a patch around each of the source and target pixels. Then, every patch is transformed in the third step, generating a new feature for each one of them. This step consists of 2 TCA stages and a feature reduction stage. In the last step, all the new features for each pixel in the source dataset are used to train a SVM classifier

that finally performs the classification of each pixel of the target dataset based on its features.

This approach has two advantages. First TCANet does not require training based on backpropagation as the filter coefficients are computed based on TCA from the input image, and therefore the computational cost is reduced. Second, DA is performed on the fly since TCA, in addition to performing dimensionality reduction, obtains components that minimize the difference in the data distributions coresponding to the source and target images. The classification accuracy evaluated using two standard datasets obtains competitive results with respect to other more complex deep learning methods for domain adaptation.

4. *Two GPU implementations of a spatial-spectral supervised classification scheme based on CNNs* were proposed, analyzed and compared.

   The proposed classification scheme, named HYCNN comprises four main steps: dimensionality reduction, patch extraction, convolution filters and classification. The first and second steps are common for the different versions of the scheme. For the first step a PCA algorithm is applied to extract the main features. Then, a patch extraction around each pixel to take the spatial information into account is carried out. The last two steps make up the forward-backward learning process: one convolutional layer processes the spectral information, and fully connected layers perform the classification. These last two steps are carried out using two different deep learning libraries: Caffe and Caffe plus cuDNN. As a result, Caffe-HYCNN and cuDNN-HYCNN are defined, being the last one the most efficient.

   In order to achieve an efficient GPU projection, different techniques and optimizations have been applied. The classification accuracy is higher for CuDNN-HYCNN and increased when an additional convolutional layer is considered, achieving values of up to 98.15% for the Pavia Centre dataset. Regarding execution times, speedups of up to 74.22x for Pavia University with respect to HYCNN in CPU are achieved.

5. *A first version of HypeRvieW, a licence-free desktop application for the analysis and classification of hyperspectral remote-sensing images* has been produced out as part of this thesis.

The objective of proposing the tool has been to provide capabilities for including the user's own developments related to remote sensing processing for land-cover applications. The application provides capabilities mainly related to data exploration and analysis in order to numerically and visually inspect the characteristics of the images, and also related to processing: reference data generation, segmentation, and supervised classification of the images, among others.

Regarding data exploration, the user can analyse details of the images inspecting them band by band, calculating some spectral indices, performing zoom, analysing the spectrum for each pixel, or even rotating, cropping or scaling the images. The proposed software also allows to create and edit the reference data for an image that is required for its supervised classification. To this end, the user can select pixels that will be assigned to a class, individually, defining polygons to group different pixels, or selecting segments previously computed by a segmentation algorithm.

Regarding the classification capabilities, different supervised spatial-spectral classification schemes can be applied by using chains each consisting of a different ordering of configurable stages: preprocessing, spatial processing, pixel-wise classification, combination, and post-processing.

The modular structure provides the user with the flexibility to create multiple scenarios and compare the classification results either visually, in terms of accuracy, or in terms of execution time. New algorithms can be included in a simple way. Indeed, this tool has been the base for introducing different algorithms proposed by the research team during the last few years, and used for learning and demonstration purposes in research, industrial, and even motivational and educational environments.

Several real hyperspectral and multispectral datasets (urban, agricultural and corresponding to riber basins) taken by the ROSIS-03, the AVIRIS and the MicaSense RedEdge sensors were used for the evaluation of the schemes and and the tool proposed in this thesis. In all the cases, the proposed schemes were compared with similar schemes under the same conditions, regarding the number of training samples and the size of the scenes.

Different hardware systems have been used in the experiments. For the sequential and the OpenMP implementations three systems were considered. The first one is a quad-core Intel i5-6600 at 3.3GHz and 32GB of RAM. The second one is quad-core Intel Xeon E5-2609v2 at 2.5 GHz and 16 GB of RAM. Finally, the third one is a quad-core Intel Core2 Quad Q9450

at 2.66 GHz and 6 GB of RAM. Two different GPUs corresponding to the Pascal and Kepler generations of CUDA capable devices were also used in the experiments. These are a Pascal NVIDIA GeForce GTX 1070 with 15 SMs and 128 CUDA cores each and a NVIDIA GeForce GTX Titan with 14 SMs and 192 CUDA cores each.

The results presented in this thesis show that the main objectives have been achieved. Different contributions such as the proposal of different classification algorithms based on neural networks and deep learning for their computation on GPU have been made. These contributions lead to the proposal of a domain adaptation technique improving the efficiency of those in the literature.

Future Work

Several possible lines for future work regarding DA for classification could be mentioned. New machine learning methods that learn from a source data distribution how to improve the performance of a classification model on a different data distribution need to be proposed. The method proposed in this thesis could be improved by improving the different stages, for example, the feature reduction stage, or trying different techniques for computing the filters of the network. As a further step, proposing methods based on the reconstruction-based DA approach seems interesting. It assumes that the data reconstruction of the source or target samples can be helpful for improving the performance of DA. The new approaches could be based on SAEs or adversarial networks. As a central requirement they should be efficiently computed on a variety of computing platforms as required by the new standard applications.

USC
UNIVERSIDADE
DE SANTIAGO
DE COMPOSTELA

Bibliography

[1] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 44–57, 2002.

[2] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 354–379, 2012.

[3] D. Manolakis and G. Shaw, "Detection algorithms for hyperspectral imaging applications," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 29–43, 2002.

[4] J. López-Fandiño, A. S. Garea, D. B. Heras, and F. Argüello, "Stacked autoencoders for multiclass change detection in hyperspectral images," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 1906–1909.

[5] J. A. Benediktsson, M. Pesaresi, and K. Amason, "Classification and feature extraction for remote sensing images from urban areas based on morphological transformations," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 41, no. 9, pp. 1940–1949, 2003.

[6] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1349–1362, 2015.

[7] L. Bruzzone and B. Demir, "A review of modern approaches to classification of remote sensing data," *Land Use and Land Cover Mapping in Europe*, pp. 127–143, 2014.

[8] A. E. Maxwell, T. A. Warner, and F. Fang, "Implementation of machine-learning classification in remote sensing: An applied review," *International Journal of Remote Sensing*, vol. 39, no. 9, pp. 2784–2817, 2018.

[9] R. R. Jensen, P. J. Hardin, and G. Yu, "Artificial neural networks and remote sensing," *Geography Compass*, vol. 3, no. 2, pp. 630–646, 2009.

[10] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, p. e00938, 2018.

[11] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, vol. 2015, 2015.

[12] J. Yue, W. Zhao, S. Mao, and H. Liu, "Spectral–spatial classification of hyperspectral images using deep convolutional neural networks," *Remote Sensing Letters*, vol. 6, no. 6, pp. 468–477, 2015.

[13] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, 2016.

[14] W. F. Schmidt, M. A. Kraaijveld, R. P. Duin *et al.*, "Feed forward neural networks with random weights," in *International Conference on Pattern Recognition*. IEEE COMPUTER SOCIETY PRESS, 1992, pp. 1–1.

[15] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[16] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 2, pp. 513–529, 2012.

[17] J. Zabalza, J. Ren, J. Zheng, H. Zhao, C. Qing, Z. Yang, P. Du, and S. Marshall, "Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging," *Neurocomputing*, vol. 185, pp. 1–10, 2016.

[18] M. Dalla Mura, A. Villa, J. A. Benediktsson, J. Chanussot, and L. Bruzzone, "Classification of hyperspectral images by using extended morphological attribute profiles

and independent component analysis," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 3, pp. 542–546, 2010.

[19] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[20] C. O. S. Sorzano, J. Vargas, and A. P. Montano, "A survey of dimensionality reduction techniques," *arXiv preprint arXiv:1403.2877*, 2014.

[21] V. Dey, Y. Zhang, and M. Zhong, *A review on image segmentation techniques with remote sensing perspective*. na Vienna, Austria, 2010, vol. 38.

[22] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 43, no. 3, pp. 480–491, 2005.

[23] P. Quesada-Barriuso, F. Argüello, and D. B. Heras, "Spectral–spatial classification of hyperspectral images using wavelets and extended morphological profiles," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (J-STARS)*, vol. 7, no. 4, pp. 1177–1185, 2014.

[24] X. Xie, "A review of recent advances in surface defect detection using texture analysis techniques," *ELCVIA: Electronic Letters on Computer Vision and Image Analysis*, pp. 1–22, 2008.

[25] W. H. Nailon, "Texture analysis methods for medical image characterisation," *Biomedical Imaging*, vol. 75, p. 100, 2010.

[26] J. Senthilnath, M. Kandukuri, A. Dokania, and K. Ramesh, "Application of UAV imaging platform for vegetation analysis based on spectral-spatial methods," *Computers and Electronics in Agriculture*, vol. 140, pp. 8 – 24, 2017.

[27] S. R. Rey, D. B. Blanco, and F. Argüello, "Texture extraction techniques for the classification of vegetation species in hyperspectral imagery: Bag of words approach based on superpixels," *Remote Sensing*, vol. 12, no. 16, p. 2633, 2020.

[28] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *European Conference on Computer Vision*. Springer, 2010, pp. 143–156.

[29] J. Macqueen, "Some methods for classification and analysis of multivariate observations," in *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[30] D. A. Clausi, "K-means iterative Fisher unsupervised clustering algorithm applied to image texture segmentation," *Pattern Recognition*, vol. 35, no. 9, pp. 1959–1972, 2002.

[31] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 3304–3311.

[32] L. Fang, S. Li, X. Kang, and J. A. Benediktsson, "Spectral–spatial classification of hyperspectral images with a superpixel-based discriminative sparse model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 8, pp. 4186–4201, 2015.

[33] X. Zhang, S. E. Chew, Z. Xu, and N. D. Cahill, "SLIC superpixels for efficient graph-based dimensionality reduction of hyperspectral imagery," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXI*, vol. 9472. International Society for Optics and Photonics, 2015, p. 947209.

[34] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[35] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, 2020.

[36] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 513–520.

[37] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[38] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?" *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017–5032, 2015.

[39] Berkeley AI Research, "Caffe website," https://caffe.berkeleyvision.org/, [Online; accessed April 20, 2021].

[40] B. Fulkerson and S. Soatto, "Really quick shift: Image segmentation on a GPU," in *Trends and Topics in Computer Vision*. Springer, 2010, pp. 350–358.

[41] Y. Zhang, X. Feng, and X. Le, "Segmentation on multispectral remote sensing image using watershed transformation," in *Image and Signal Processing, 2008. CISP'08. Congress on*, vol. 4. IEEE, 2008, pp. 773–777.

[42] Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," *Pattern Recognition*, vol. 43, no. 7, pp. 2367–2379, 2010.

[43] W. Pan, K. Qin, and Y. Chen, "An adaptable-multilayer fractional Fourier transform approach for image registration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 400–414, 2008.

[44] B. Kröse, B. Krose, P. van der Smagt, and P. Smagt, "An introduction to neural networks," 1993.

[45] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological Review*, vol. 65, no. 6, p. 386, 1958.

[46] C. C. Aggarwal *et al.*, "Neural networks and deep learning," *Springer*, vol. 10, pp. 978–3, 2018.

[47] K. Suzuki, *Artificial neural networks: methodological advances and biomedical applications*. BoD–Books on Demand, 2011.

[48] V. Sharma, S. Rai, and A. Dev, "A comprehensive study of artificial neural networks," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 10, 2012.

[49] L. Bottou, "Stochastic gradient learning in neural networks," *Proceedings of Neuro-Nımes*, vol. 91, no. 8, p. 12, 1991.

[50] K. Gurney, *An introduction to neural networks*. CRC press, 2014.

[51] W. Cao, X. Wang, Z. Ming, and J. Gao, "A review on neural networks with random weights," *Neurocomputing*, vol. 275, pp. 278–287, 2018.

[52] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, vol. 2. IEEE, 2004, pp. 985–990.

[53] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 11, pp. 3804–3814, 2008.

[54] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 652–675, 2012.

[55] M. Pesaresi and J. A. Benediktsson, "A new approach for the morphological segmentation of high–resolution satellite imagery," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 39, no. 2, pp. 309–320, 2001.

[56] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone, "Morphological attribute profiles for the analysis of very high resolution images," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 48, no. 10, pp. 3747–3762, 2010.

[57] P. Ghamisi, R. Souza, J. A. Benediktsson, X. X. Zhu, L. Rittner, and R. A. Lotufo, "Extinction profiles for the classification of remote sensing data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 5631–5645, 2016.

[58] S.-C. Zhu, C.-E. Guo, Y. Wang, and Z. Xu, "What are textons?" *International Journal of Computer Vision*, vol. 62, no. 1, pp. 121–143, 2005.

[59] B. Julesz, "Textons, the elements of texture perception, and their interactions," *Nature*, vol. 290, no. 5802, pp. 91–97, 1981.

[60] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision, ECCV*, vol. 1, no. 1-22. Prague, 2004, pp. 1–2.

[61] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[62] J. Ritchie, "Climate change and vegetation response," *Vegetatio*, vol. 67, no. 2, pp. 65–74, 1986.

[63] V. Lawley, M. Lewis, K. Clarke, and B. Ostendorf, "Site-based and remote sensing methods for monitoring indicators of vegetation condition: An Australian review," *Ecological Indicators*, vol. 60, pp. 1273–1283, 2016.

[64] S. Li, W. Liang, B. Fu, Y. Lü, S. Fu, S. Wang, and H. Su, "Vegetation changes in recent large-scale ecological restoration projects and subsequent impact on water resources in China's Loess Plateau," *Science of the Total Environment*, vol. 569, pp. 1032–1039, 2016.

[65] P. S. Thenkabail and J. G. Lyon, *Hyperspectral remote sensing of vegetation*. CRC press, 2016.

[66] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6690–6709, 2019.

[67] D. J. Lary, A. H. Alavi, A. H. Gandomi, and A. L. Walker, "Machine learning in geosciences and remote sensing," *Geoscience Frontiers*, vol. 7, no. 1, pp. 3–10, 2016.

[68] B. Pan, Z. Shi, and X. Xu, "R-VCANet: A new deep-learning-based hyperspectral image classification method," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 5, pp. 1975–1986, 2017.

[69] J. Li, X. Zhao, Y. Li, Q. Du, B. Xi, and J. Hu, "Classification of hyperspectral imagery using a new fully convolutional neural network," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 292–296, 2018.

[70] X. Yang, Y. Ye, X. Li, R. Y. Lau, X. Zhang, and X. Huang, "Hyperspectral image classification with deep learning models," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 9, pp. 5408–5423, 2018.

[71] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017.

[72] Y. Li, H. Zhang, X. Xue, Y. Jiang, and Q. Shen, "Deep learning for remote sensing image classification: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 6, p. e1264, 2018.

[73] D. Tuia, C. Persello, and L. Bruzzone, "Domain adaptation for the classification of remote sensing data: An overview of recent advances," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 41–57, 2016.

[74] H. Venkateswara, S. Chakraborty, and S. Panchanathan, "Deep-learning systems for domain adaptation in computer vision: Learning transferable feature representations," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 117–129, 2017.

[75] M. E. Yuksel, N. S. Basturk, H. Badem, A. Caliskan, and A. Basturk, "Classification of high resolution hyperspectral remote sensing data using deep neural networks," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 4, pp. 2273–2285, 2018.

[76] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 645–657, 2016.

[77] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2381–2392, 2015.

[78] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[79] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[80] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Advances in Neural Information Processing Systems*, 2007, pp. 137–144.

[81] S. Song, H. Yu, Z. Miao, Q. Zhang, Y. Lin, and S. Wang, "Domain adaptation for convolutional neural networks-based remote sensing scene classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 8, pp. 1324–1328, 2019.

[82] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*. ACM, 2008, pp. 1096–1103.

[83] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*. Omnipress, 2011, pp. 833–840.

[84] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, and M. Marchand, "Domain-adversarial neural networks," *arXiv preprint arXiv:1412.4446*, 2014.

[85] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[86] A. Elshamli, G. W. Taylor, A. Berg, and S. Areibi, "Domain adaptation using representation learning for the classification of remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 9, pp. 4198–4209, 2017.

[87] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," *arXiv preprint arXiv:1508.00092*, 2015.

[88] Z. Huang, W. Xue, Q. Mao, and Y. Zhan, "Unsupervised domain adaptation for speech emotion recognition using PCANet," *Multimedia Tools and Applications*, vol. 76, no. 5, pp. 6785–6799, 2017.

[89] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[90] Google, "TensorFlow website," https://www.tensorflow.org/, [Online; accessed January 5, 2021].

[91] Facebook's AI Research lab (FAIR), "PyTorch website," https://pytorch.org/, [Online; accessed January 5, 2021].

[92] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM International Conference on Multimedia*. ACM, 2014, pp. 675–678.

[93] F. Chollet, "Keras website," https://keras.io/, [Online; accessed January 5, 2021].

[94] NVIDIA, "CUDA toolkit website," https://developer.nvidia.com/cuda-toolkit, [Online; accessed January 5, 2021].

[95] A. S. Garea, D. B. Heras, and F. Argüello, "GPU classification of remote-sensing images using kernel ELM and extended morphological profiles," *International Journal of Remote Sensing*, vol. 37, no. 24, pp. 5918–5935, 2016.

[96] A. S. Garea, D. B. Heras, and F. Argüello, "Clasificación de imágenes de teledetección mediante ELM kernel y perfiles morfológicos en GPU," in *Jornadas SARTECO 2016*. Ediciones Universidad de Salamanca, 2016, pp. 81–89.

[97] P. G. Bascoy, A. S. Garea, D. B. Heras, F. Argüello, and A. Ordóñez, "Texture-based analysis of hydrographical basins with multispectral imagery," in *Remote Sensing for Agriculture, Ecosystems, and Hydrology XXI*, C. M. U. Neale and A. Maltese, Eds., vol. 11149, International Society for Optics and Photonics. SPIE, 2019, pp. 225 – 234.

[98] A. S. Garea, D. B. Heras, and F. Argüello, "Caffe CNN-based classification of hyperspectral images on GPU," *The Journal of Supercomputing*, vol. 75, no. 3, pp. 1065–1077, 2019.

[99] A. S. Garea, D. B. Heras, and F. Argüello, "GPU classification for hyperspectral images based on convolutional neural networks," in *Computational and Mathematical Methods in Science and Engineering*, J. Vigo-Aguiar, Ed., 2017, pp. 912–923.

[100] A. S. Garea, D. B. Heras, and F. Argüello, "TCANet for domain adaptation of hyperspectral images," *Remote Sensing*, vol. 11, no. 19, p. 2289, 2019.

[101] A. S. Garea, Á. Ordóñez, D. B. Heras, and F. Argüello, "HypeRvieW: an open source desktop application for hyperspectral remote-sensing data processing," *International Journal of Remote Sensing*, vol. 37, no. 23, pp. 5533–5550, 2016.

[102] A. S. Garea, D. B. Heras, and F. Arguëllo, "An open source desktop application for classification of remote sensing data," in *2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 1. IEEE, 2015, pp. 316–321.

[103] OpenMP Architecture Review Board, "OpenMP website," https://www.openmp.org/, [Online; accessed March 8, 2021].

[104] NVIDIA, "Fermi GPU family whitepaper," https://www.nvidia.es/content/PDF/fermi_white_papers/NVIDIA_Fermi_Compute_Architecture_Whitepaper.pdf, [Online; accessed April 15, 2021].

[105] Wikipedia, "GeForce 8 series," https://en.wikipedia.org/wiki/GeForce_8_series, [Online; accessed January 12, 2021].

[106] NVIDIA, "GeForce RTX 30 series," https://www.nvidia.com/en-gb/geforce/graphics-cards/30-series/, [Online; accessed April 15, 2021].

[107] M. S. Serpa, F. B. Moreira, P. O. Navaux, E. H. Cruz, M. Diener, D. Griebler, and L. G. Fernandes, "Memory performance and bottlenecks in multicore and GPU architectures," in *2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*. IEEE, 2019, pp. 233–236.

[108] NVIDIA, "CUDA C++ best practices guide," https://docs.nvidia.com/cuda/cuda-c-best-practices-guide/index.html, [Online; accessed April 23, 2021].

[109] NVIDIA, "Kepler GPU family whitepaper," https://www.microway.com/download/whitepaper/NVIDIA_Kepler_GK110_GK210_Architecture_Whitepaper.pdf, [Online; accessed April 15, 2021].

[110] NVIDIA, "Pascal GPU family whitepaper," https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf, [Online; accessed April 15, 2021].

[111] GNU Project, "GCC website," https://gcc.gnu.org/, [Online; accessed April 23, 2021].

[112] Python Software foundation, "Python website," https://www.python.org/, [Online; accessed January 5, 2021].

[113] NVIDIA, "CUDA toolkit documentation: CUBLAS," http://docs.nvidia.com/cuda/cublas/index.html, [Online; accessed February 5, 2021].

[114] NVIDIA, "CULA tools website," http://www.culatools.com/cula_dense_programmers_guide/, [Online; accessed February 11, 2021].

[115] OpenBLAS, "OpenBLAS website," http://www.openblas.net/, [Online; accessed January 17, 2021].

[116] NVIDIA, "cuDNN website," https://developer.nvidia.com/cudnn, [Online; accessed February 2, 2021].

[117] E. R. Frank Warmerdam *et al.*, "GDAL website," https://gdal.org/, [Online; accessed April 23, 2021].

[118] OpenCV Team, "OpenCV website," https://opencv.org/, [Online; accessed April 23, 2021].

[119] A. J. Viera, J. M. Garrett *et al.*, "Understanding interobserver agreement: the kappa statistic," *Fam Med*, vol. 37, no. 5, pp. 360–363, 2005.

[120] G. M. Foody, "Thematic map comparison," *Photogrammetric Engineering & Remote Sensing*, vol. 70, no. 5, pp. 627–633, 2004.

[121] J. De Leeuw, H. Jia, L. Yang, X. Liu, K. Schmidt, and A. Skidmore, "Comparing accuracy assessments to infer superiority of image classification methods," *International Journal of Remote Sensing*, vol. 27, no. 1, pp. 223–232, 2006.

[122] R. G. J. Pontius and M. Millones, "Death to kappa: birth of quantity disagreement and allocation disagreement for accuracy assessment." *International Journal of Remote Sensing*, vol. 32, no. 15, pp. 4407–4429, 2011.

[123] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[124] G. Vane, "First results from the airborne visible/infrared imaging spectrometer (AVIRIS)," in *Imaging Spectroscopy II*, vol. 834. International Society for Optics and Photonics, 1987, pp. 166–175.

[125] R. O. Green, J. E. Conel, M. Helmlinger, J. van den Bosch, C. Chovit, and T. Chrien, "Inflight calibration of AVIRIS in 1992 and 1993," *Summaries of the Fourth Annual JPL Airboene Geoscience Workshop*, vol. 1, 1993.

[126] R. O. Green, M. L. Eastwood, C. M. Sarture, T. G. Chrien, M. Aronsson, B. J. Chippendale, J. A. Faust, B. E. Pavri, C. J. Chovit, M. Solis *et al.*, "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sensing of Environment*, vol. 65, no. 3, pp. 227–248, 1998.

[127] A. Müller, P. Gege, and T. Cocks, "The airborne imaging spectrometers used in DAISEX," in *The Digital Airborne Spectrometer Experiment (DAISEX)*, vol. 499, 2001, p. 3.

[128] S. Holzwarth, A. Muller, M. Habermeyer, R. Richter, A. Hausold, S. Thiemann, and P. Strobl, "HySens-DAIS 7915/ROSIS imaging spectrometers at DLR," in *Proceedings of the 3rd EARSeL Workshop on Imaging Spectroscopy*, 2003, pp. 3–14.

[129] K. Lenhard, "Determination of combined measurement uncertainty via Monte Carlo analysis for the imaging spectrometer ROSIS," *Applied Optics*, vol. 51, no. 18, pp. 4065–4072, 2012.

[130] "ROSIS University of Pavia dataset, 2002," http://www.ehu.es/ccwintco/uploads/e/ee/PaviaU.mat (103–band hyperspectral image) and http://www.ehu.es/ccwintco/uploads/5/50/PaviaU_gt.mat (ground truth), [Online; accessed January 10, 2021].

[131] F. Argüello and D. B. Heras, "ELM-based spectral-spatial classification of hyperspectral images using extended morphological profiles and composite feature mappings," *International Journal of Remote Sensing*, vol. 36, no. 2, pp. 645–664, 2015.

[132] "AVIRIS Indian Pine test site 3 dataset (2x2 mile portion of northwest Tippecanoe County, Indiana), 1992," ftp://ftp.ecn.purdue.edu/biehl/MultiSpec/92AV3C.tif.

zip (220–band hyperspectral image) and ftp://ftp.ecn.purdue.edu/biehl/MultiSpec/ThyFiles.zip (ground truth), [Online; accessed January 10, 2021].

[133] "AVIRIS Salinas dataset," http://www.ehu.eus/ccwintco/uploads/a/a3/Salinas_corrected.mat (200–band hyperspectral image) and http://www.ehu.eus/ccwintco/uploads/f/fa/Salinas_gt.mat (ground truth), [Online; accessed January 10, 2021].

[134] J. López-Fandiño, P. Quesada-Barriuso, D. B. Heras, and F. Argüello, "Efficient ELM-based techniques for the classification of hyperspectral remote sensing images on commodity GPUs," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 8, no. 6, pp. 2884–2893, 2015.

[135] "ROSIS Pavia centre dataset, 2002," http://www.ehu.es/ccwintco/uploads/e/e3/Pavia.mat (102–band hyperspectral image) and http://www.ehu.es/ccwintco/uploads/5/53/Pavia_gt.mat (ground truth), [Online; accessed January 10, 2021].

[136] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe, "220 band AVIRIS hyperspectral image dataset: June 12, 1992 Indian Pine test site 3," https://purr.purdue.edu/publications/1947/1, [Online; accessed January 10, 2021].

[137] H. Sajjad and P. Kumar, "Future challenges and perspective of remote sensing technology," in *Applications and Challenges of Geospatial Technology*. Springer, 2019, pp. 275–277.

[138] C. D. Lippitt and S. Zhang, "The impact of small unmanned airborne platforms on passive optical remote sensing: A conceptual perspective," *International journal of Remote Sensing*, vol. 39, no. 15-16, pp. 4852–4868, 2018.

[139] S. Gopal and C. Woodcock, "Remote sensing of forest change using artificial neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 2, pp. 398–404, 1996.

[140] J. C. A. Barata and M. S. Hussein, "The Moore–Penrose pseudoinverse: A tutorial review of the theory," *Brazilian Journal of Physics*, vol. 42, no. 1-2, pp. 146–165, 2012.

[141] P. L. Bartlett, "The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 525–536, 1998.

[142] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 6, pp. 1351–1362, 2005.

[143] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[144] J. Chorowski, J. Wang, and J. M. Zurada, "Review and performance comparison of SVM-and ELM-based classifiers," *Neurocomputing*, vol. 128, pp. 507–516, 2014.

[145] M. Bucurica, R. Dogaru, and I. Dogaru, "A comparison of extreme learning machine and support vector machine classifiers," in *2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE, 2015, pp. 471–474.

[146] L. Zhang, D. Zhang, and F. Tian, "SVM and ELM: Who wins? object recognition with deep convolutional features from ImageNet," in *Proceedings of ELM-2015 Volume 1*. Springer, 2016, pp. 249–263.

[147] M. Andrecut, "Parallel GPU implementation of iterative PCA algorithms," *Journal of Computational Biology*, vol. 16, no. 11, pp. 1593–1599, 2009.

[148] P. Soille, *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013.

[149] A. Bannari, D. Morin, F. Bonn, and A. Huete, "A review of vegetation indices," *Remote Sensing Reviews*, vol. 13, no. 1-2, pp. 95–120, 1995.

[150] J. Xue and B. Su, "Significant remote sensing vegetation indices: A review of developments and applications," *Journal of Sensors*, vol. 2017, 2017.

[151] Y. Fu, C. Zhao, J. Wang, X. Jia, G. Yang, X. Song, and H. Feng, "An improved combination of spectral and spatial features for vegetation classification in hyperspectral images," *Remote Sensing*, vol. 9, no. 3, 2017.

[152] J. Oldeland, A. Große-Stoltenberg, L. Naftal, and B. Strohbach, "The potential of UAV derived image features for discriminating savannah tree species," in *The roles of remote sensing in nature conservation*. Springer, 2017, pp. 183–201.

[153] D. Harris, J. Vlok, and A. van Niekerk, "Regional mapping of spekboom canopy cover using very high resolution aerial imagery," *Journal of Applied Remote Sensing*, vol. 12, no. 4, p. 046022, 2018.

[154] W. A. Dominik, "Exploiting the redundancy of multiple overlapping aerial images for dense image matching based digital surface model generation," *Remote Sensing*, vol. 9, no. 5, p. 490, 2017.

[155] K. Osińska-Skotak, K. Bakuła, Ł. Jełowicki, and A. Podkowa, "Using canopy height model obtained with dense image matching of archival photogrammetric datasets in area analysis of secondary succession," *Remote Sensing*, vol. 11, no. 18, p. 2182, 2019.

[156] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3606–3613.

[157] X. Zhang, Y. Sun, K. Shang, L. Zhang, and S. Wang, "Crop classification based on feature band set construction and object-oriented approach using hyperspectral images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 9, pp. 4117–4128, 2016.

[158] G.-H. Kwak and N.-W. Park, "Impact of texture information on crop classification with machine learning and UAV images," *Applied Sciences*, vol. 9, no. 4, p. 643, 2019.

[159] J. Li, H. Zhang, and L. Zhang, "Efficient superpixel-level multitask joint sparse representation for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 10, pp. 5338–5351, 2015.

[160] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," *arXiv preprint arXiv:2003.05991*, 2020.

[161] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*. JMLR Workshop and Conference Proceedings, 2012, pp. 37–49.

[162] M. Tschannen, O. Bachem, and M. Lucic, "Recent advances in autoencoder-based representation learning," *arXiv preprint arXiv:1812.05069*, 2018.

[163] A. O. B. Özdemir, B. E. Gedik, and C. Y. Y. Çetin, "Hyperspectral classification using stacked autoencoders with deep learning," in *2014 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. IEEE, 2014, pp. 1–4.

[164] P. Liang, W. Shi, and X. Zhang, "Remote sensing image classification based on stacked denoising autoencoder," *Remote Sensing*, vol. 10, no. 1, p. 16, 2018.

[165] A. Belwalkar, A. Nath, and O. Dikshit, "Spectral-spatial classification of hyperspectral remote sensing images using variational autoencoder and convolution neural network." *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2018.

[166] B. Ayhan and C. Kwan, "Application of deep belief network to land cover classification using hyperspectral images," in *International Symposium on Neural Networks*. Springer, 2017, pp. 269–276.

[167] C. Li, Y. Wang, X. Zhang, H. Gao, Y. Yang, and J. Wang, "Deep belief network for spectral–spatial classification of hyperspectral remote sensor data," *Sensors*, vol. 19, no. 1, p. 204, 2019.

[168] M. Midhun, S. R. Nair, V. N. Prabhakar, and S. S. Kumar, "Deep model for classification of hyperspectral image using restricted boltzmann machine," in *Proceedings of the 2014 International Conference on Interdisciplinary Advances in Applied Computing*, 2014, pp. 1–7.

[169] Y. Zhan, D. Hu, Y. Wang, and X. Yu, "Semisupervised hyperspectral image classification based on generative adversarial networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 212–216, 2017.

[170] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Generative adversarial networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 9, pp. 5046–5063, 2018.

[171] J. Feng, H. Yu, L. Wang, X. Cao, X. Zhang, and L. Jiao, "Classification of hyperspectral images based on multiclass spatial–spectral generative adversarial networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5329–5343, 2019.

[172] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[173] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*. Ieee, 2017, pp. 1–6.

[174] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International Journal of Remote Sensing*, vol. 28, no. 5, pp. 823–870, 2007.

[175] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset shift in machine learning*. The MIT Press, 2009.

[176] G. Jia, A. Hueni, M. E. Schaepman, and H. Zhao, "Detection and correction of spectral shift effects for the airborne prism experiment," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 11, pp. 6666–6679, 2017.

[177] R. G. Congalton, "A review of assessing the accuracy of classifications of remotely sensed data," *Remote Sensing of Environment*, vol. 37, no. 1, pp. 35–46, 1991.

[178] L. Bruzzone and C. Persello, "A novel approach to the selection of spatially invariant features for the classification of hyperspectral images with improved generalization capability," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 9, pp. 3180–3191, 2009.

[179] E. Izquierdo-Verdiguier, V. Laparra, L. Gomez-Chova, and G. Camps-Valls, "Encoding invariances in remote sensing image classification with SVM," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 5, pp. 981–985, 2013.

[180] C. Persello and L. Bruzzone, "Kernel-based domain-invariant feature selection in hyperspectral images for transfer learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 5, pp. 2615–2626, 2015.

[181] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A DASVM classification technique and a circular validation strategy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 770–787, 2010.

[182] W. Kim and M. M. Crawford, "Adaptive classification for hyperspectral image data using manifold regularization kernel machines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4110–4121, 2010.

[183] D. Tuia, E. Pasolli, and W. J. Emery, "Using active learning to adapt remote sensing image classifiers," *Remote Sensing of Environment*, vol. 115, no. 9, pp. 2232–2242, 2011.

[184] C. Persello and L. Bruzzone, "Active learning for domain adaptation in the supervised classification of remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 11, pp. 4468–4483, 2012.

[185] C. Persello, "Interactive domain adaptation for the classification of remote sensing images using active learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 4, pp. 736–740, 2013.

[186] GRASS GIS, "GRASS GIS website," https://grass.osgeo.org, [Online; accessed January 18, 2021].

[187] Supported by several partnerships, "QGIS website," https://www.qgis.org, [Online; accessed January 18, 2021].

[188] gvGIS Institution, "gvGIS website," http://www.gvsig.com, [Online; accessed January 18, 2021].

[189] The University of Chicago, "GeoDa website," https://geodacenter.github.io, [Online; accessed January 18, 2021].

[190] L3Harris Geospatial, "ENVI website," https://www.l3harrisgeospatial.com/Software-Technology/ENVI, [Online; accessed January 18, 2021].

[191] Trimble, "eCognition website," http://www.ecognition.com, [Online; accessed January 18, 2021].

[192] Esri, "ArcGIS website," https://www.arcgis.com, [Online; accessed January 18, 2021].

[193] A. S. Garea, Á. Ordóñez, D. B. Heras, and F. Argüello, “HypeRvieW website,” https://wiki.citius.usc.es/hiperespectral:hyperview, [Online; accessed January 18, 2021].

List of Figures

USC
UNIVERSIDADE
DE SANTIAGO
DE COMPOSTELA

List of Tables