

**UNIVERSIDADE DE SANTIAGO DE
COMPOSTELA**



ESCOLA TÉCNICA SUPERIOR DE ENXEÑARÍA

**Adaptación efectiva de redes xerativas
antagónicas para o procesado de imaxes
de teledetección multidimensionais**

Autor/a:
Antón Gómez López

Titores:
Francisco Santiago Argüello Pedreira
Dora Blanco Heras

Grao en Enxeñaría Informática

Xullo 2024

Traballo de Fin de Grao presentado na Escola Técnica Superior de Enxeñaría
da Universidade de Santiago de Compostela para a obtención do Grao en
Enxeñaría Informática



D. Francisco Santiago Argüello Pedreira, Profesor do Departamento de Electrónica e Computación da Universidade de Santiago de Compostela, e **D. Dora Blanco Heras**, Profesora do Departamento de Electrónica e Computación da Universidade de Santiago de Compostela,

INFORMAN:

Que a presente memoria, titulada *Adaptación efectiva de redes xerativas antagónicas para o procesado de imaxes de teledetección multidimensionais*, presentada por **D. Antón Gómez López** para superar os créditos correspondentes ao Traballo de Fin de Grao da titulación de Grao en Enxeñaría Informática, realizouse baixo nosa titoría no Departamento de Electrónica e Computación da Universidade de Santiago de Compostela.

E para que así conste aos efectos oportunos, expiden o presente informe en Santiago de Compostela, a (Data):

Titor/a,

Cotitor/a,

Alumno/a,

(Nome do titor/a) (Nome do cotitor/a) (Nome do alumn/a)

Resumo

Nos últimos anos propuxéronse diversos modelos de clasificación de aprendizaxe profunda que se sumaron aos existentes métodos de clasificación de imaxes multiespectrais de teledetección. Dentro deste marco, neste traballo, adaptouse unha rede xerativa antagónica condicionada, baseada na arquitectura do modelo StyleGAN2, ao problema de clasificación de imaxes multiespectrais de teledetección. Posteriormente, realizouse un estudo sobre a capacidade da rede, tanto para xerar, como para clasificar imaxes multiespectrais de alta resolución espacial correspondentes a ríos galegos.

Finalmente, comparáronse os resultados acadados con este modelo de clasificación e outros modelos empregados en problemas de teledetección multiespectral. Observouse que a StyleGAN2 condicionada acada resultados próximos aos doutros esquemas de clasificación que non empregan conxuntos de datos xerados, como as redes neuronais convolucionais, mais queda por debaixo de métodos especialmente deseñados para xerar mostras para problemas de clasificación con clases desbalanceadas, como é a ResBaGAN.

Palabras chave: clasificación, GAN, imaxe multiespectral, StyleGAN2, teledetección.

Resumen

En los últimos años se han propuesto diversos modelos de clasificación de aprendizaje profundo que se sumaron a los existentes métodos de clasificación de imágenes multiespectrales de teledetección. Dentro de este marco, en este trabajo, se adaptó una red generativa antagónica condicionada, basada en la arquitectura del modelo StyleGAN2 al problema de clasificación de imágenes multiespectrales de teledetección. Posteriormente, se realizó un estudio sobre la capacidad de la red, tanto para generar, como para clasificar imágenes multiespectrales de alta resolución espacial correspondientes a ríos gallegos.

Finalmente, se compararon los resultados alcanzados con este modelo de clasificación y otros modelos empleados en problemas de teledetección multiespectral. Se observó que la StyleGAN2 condicionada alcanza resultados cercanos a otros esquemas de clasificación que no emplean conjuntos de datos generados, como las redes neuronales convolucionales, pero queda por debajo de métodos especialmente diseñados para generar muestras para problemas de clasificación con clases desbalanceadas, como es la ResBaGAN.

Palabras clave: clasificación, GAN, imagen multiespectral, StyleGAN2, teledetección.

Abstract

In recent years, various deep learning classification models have been proposed, adding to the existing methods for classifying multispectral remote sensing images. Within this framework, in this work, a conditional generative adversarial network (GAN), based on the StyleGAN2 model architecture, was adapted to the problem of classifying multispectral remote sensing images. Subsequently, a study was conducted on the network's capability to both generate and classify high spatial resolution multispectral images corresponding to Galician rivers.

Finally, the results achieved with this classification model were compared to other models used in multispectral remote sensing problems. It was observed that the conditional StyleGAN2 achieves results close to other classification schemes that do not use generated datasets, such as convolutional neural networks, but falls short of methods specifically designed to generate samples for classification problems with imbalanced classes, such as ResBaGAN.

Keywords: classification, GAN, multispectral image, StyleGAN2, remote sensing.

Siglas

AA *Average Accuracy*

cGAN *Rede Xerativa Antagónica Condicionada*

CNN *Rede Neuronal Convolucional*

CPU *Unidade de Procesamento Central*

CUDA *Arquitectura Unificada de Dispositivos de Cómputo*

DL *Aprendizaxe Profunda*

ER *Taxa de erro*

GAN *Rede Xerativa Antagónica*

GPU *Unidade de Procesamento Gráfica*

LR *Taxa de Aprendizaxe*

LR-ADA *Taxa de Aprendizaxe Adaptativa*

MSI *Imaxe Multiespectral*

OA *Overall Accuracy*

RAM *Memoria de Acceso Aleatorio*

RGB *Vermello, Verde, Azul*

SLIC *Simple Linear Iterative Clustering*

SSH *Secure Shell*

SVM *Máquina de Vectores de Soporte*

UAV *Vehículo Aéreo non Tripulado*

WP *WaterPixels*

Índice xeral

1. Introdución	1
2. Estado do coñecemento	5
3. Metodoloxía	9
3.1. Preprocesamento dos datos: segmentación	9
3.2. Modelo de clasificación	11
3.2.1. Xerador da StyleGAN	11
3.2.2. Xerador da StyleGAN2	14
3.2.3. Discriminador dunha <i>Rede Xerativa Antagónica</i> (GAN) típica	14
3.2.4. StyleGAN2 condicionada	19
3.2.5. Adaptación da StyleGAN2 condicionada para clasificación	21
4. Materiais e métricas	23
4.1. Conxuntos de datos	23
4.2. Métricas de avaliación	26
4.3. Entorno de execución	26
4.3.1. Hardware	27
4.3.2. Software	27
5. Probas	29
5.1. Plan de probas	29
5.1.1. Experimento 1: Validación do modelo de clasificación	30
5.1.2. Preprocesamento das imaxes multiespectrais	31
5.1.3. Experimento 2: Optimización de hiperparámetros	32
5.1.4. Experimento 3: Avaliación con todos os conxuntos de datos	35
5.2. Descripción dos experimentos	35
5.3. Resultados	36
6. Discusión dos resultados	43
6.1. Validación do modelo	43
6.2. Optimización de hiperparámetros	43

6.3. Comparación de resultados	44
7. Conclusións e posibles ampliacións	45
A. Manuais técnicos	47
A.1. Adquisición do código	47
A.2. Instalación de dependencias	47
A.3. Adquisición dos conxuntos de datos	48
B. Manuais de usuario	49
B.1. Repetición dos experimentos	49
B.2. Problemas frecuentes	51
C. Licenza	53

Capítulo 1

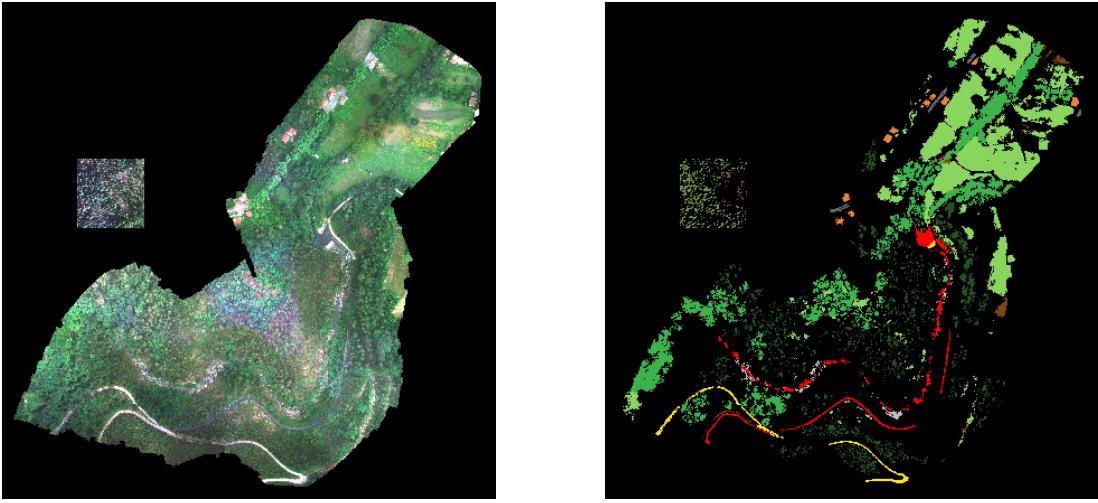
Introdución

A teledetección é a técnica que permite obter información sobre un obxecto ou fenómeno sen contacto físico directo. Utilizando sensores situados en satélites, avións ou drons, a teledetección permite recoller imaxes que son posteriormente procesadas para extraer información relevante sobre a superficie terrestre. Este método é fundamental para monitorizar a Terra, facilitando tarefas como o seguimento de construcións humanas [20], a detección de cambios na agricultura [12] ou o estudo de ecosistemas [3].

Os sensores de imaxes multiespectrais son un tipo de sensor utilizado en teledetección. Estes sensores capturan información en varias bandas espectrais, incluíndo tanto o espectro visible como partes do infravermello próximo e medio. Unha imaxe capturada por estes sensores denomínase *Imaxe Multiespectral* (MSI). Este tipo de imaxes permiten unha análise máis detallada e precisa das propiedades da superficie terrestre en comparación coas imaxes con tres bandas *Vermello, Verde, Azul* (RGB) tradicionais.

No ámbito da teledetección con imaxes MSI, a clasificación dunha imaxe consiste en asignar unha etiqueta a cada píxel desta, indicando a que clase pertence. Por exemplo, nunha imaxe de sensado remoto dunha cunca dun río, o obxectivo consiste en recoñecer os distintos tipos de especies arbóreas así como outros obxectos que poden aparecer nesta, como auga, camiños, tellados, etc. Na Figura 1.1a amósanse as bandas visibles dunha imaxe MSI da cunca do río Oitavén e na Figura 1.1b o resultado da clasificación dos píxeles desta imaxe.

Tradicionalmente, os problemas de clasificación en teledetección abordáronse mediante métodos de clasificación clásicos, por exemplo, as *Máquina de Vectores de Soporte* (SVM) [22]. Porén, nos últimos anos, a evolución dos modelos de *Aprendizaxe Profunda* (DL) permitiu que puidesen ser aplicados tamén neste



(a) Imaxe en RGB da cunca do río Oitavén.

(b) Resultado da clasificación dos píxeles da imaxe da cunca do río Oitavén.

Figura 1.1: Exemplo da clasificación dunha imaxe MSI.

tipo de problemas. A *Rede Neuronal Convolucional* (CNN) é un tipo de modelo de DL amplamente empregado en problemas de clasificación de imaxes, que se estendeu á clasificación con imaxes MSI [8, 16, 19]. As CNN son capaces de aprender patróns complexos nas imaxes sen necesidade de extraer manualmente características das imaxes.

Polo xeral, os conjuntos de datos de imaxes de sensado remoto presentan problemas de desequilibrio de clases e escaseza de datos. Para afrontar este problema, utilizanse técnicas de aumentado de datos que buscan enriquecer o conjunto de datos de adestramento, aumentándoo e mellorando a súa calidade, de xeito que sexa posible construír mellores modelos de DL con eles [21]. As técnicas de aumentado tradicionais aplican transformacións ás imaxes do conjunto de datos para obter novas imaxes que se utilizan para adestrar o modelo. Por outro lado, as técnicas de aumentado más modernas utilizan técnicas xerativas para producir imaxes dende cero, sen necesidade de transformar imaxes existentes.

Actualmente, o enfoque xerativo de datos más amplamente adoptado son as GAN [7]. Estas son un tipo de modelo de DL que consisten en dous redes neuronais, unha xeradora e unha discriminadora, que compiten entre si. A rede xeradora produce imaxes sintéticas que tratan de imitar as imaxes do conjunto de adestramento, mentres que a rede discriminadora trata de distinguir entre imaxes reais e sintéticas. Desta maneira, unha vez adestradas ambas redes, o xerador pode ser empregado para producir imaxes sintéticas e aumentar o conjunto de datos de adestramento.

Os avances na arquitectura das GAN permitiron que o discriminador poida ser utilizado tamén como clasificador. Para iso, é necesario adaptar a arquitectura da GAN de maneira que o xerador poida producir imaxes condicionadas por un vector de clase, e que o discriminador funcione como clasificador. Este tipo de GAN coñécese como *Rede Xerativa Antagónica Condicionada* (cGAN).

O modelo StyleGAN2 [10] é unha GAN capaz de xerar imaxes de alta calidade [24]. Unha das principais novedades da arquitectura StyleGAN2 e da súa predecesora, StyleGAN [9], é a capacidade de controlar as características das imaxes xeradas a través duns vectores que denominan “estilos”.

Neste traballo utilizarase unha versión condicionada da StyleGAN2 disponible en [5] para clasificar imaxes MSI. Esta rede, a pesar de estar condicionada, non permite utilizar o discriminador como clasificador. É dicir, soamente permite xerar imaxes condicionadas a un vector de clase, pero non permite asignar unha etiqueta de clase a unha imaxe de entrada. Así, o obxectivo principal deste traballo é adaptar a StyleGAN2 condicionada ao problema de clasificación de imaxes MSI. En particular, os obxectivos específicos deste traballo son:

- Introducir o problema da clasificación de imaxes MSI, explicando as etapas de preprocessado necesarias, como a segmentación en superpíxeles e a extracción de parches.
- Explicar o funcionamento dos modelos StyleGAN co obxectivo de entender e xustificar as modificacións necesarias para adaptar a StyleGAN2 condicionada ao problema de clasificación de imaxes MSI.
- Modificar o discriminador da StyleGAN2 condicionada para que poida ser utilizado como clasificador.
- Avaliar a capacidade da StyleGAN2 condicionada para clasificar imaxes MSI.

Capítulo 2

Estado do coñecemento

Nos últimos anos, as redes CNN demostraron ser moi eficaces no ámbito da clasificación de imaxes de teledetección. Por exemplo, Hamdi et al. [8] utilizaron unha CNN para detectar áreas danadas nunha área forestal en Bavaria, mentres que Morales et al. [16] empregaron unha CNN para monitorizar a deforestación da palmeira Mauritia flexuosa. Estas redes son capaces de aprender patróns complexos nas imaxes, o que as fai especialmente útiles para a clasificación de imaxes MSI.

A arquitectura típica das CNN para clasificación consiste nunha serie de capas que extraen características e unha capa densa que realiza a tarefa de clasificación a partir das características extraídas. As CNNs comezan cunha capa de entrada que recibe as imaxes en formato de matriz de píxeles. A continuación, veñen unha ou máis capas convolucionais, nas que se aplican filtros para reducir a resolución das imaxes e aumentar o número de canles. Xeralmente, para extraer características, combínase unha convolución, que aumenta as canles da imaxe, cunha operación que reduce a resolución da imaxe, como o *max-pooling* ou o *average-pooling*.

Como saída das capas convolucionais obtense un vector, que se pode interpretar como un conxunto de características aprendidas da imaxe. Deste xeito, as capas convolucionais encárganse de resumir as imaxes de entrada en vectores de características. Finalmente, unha capa densa (ou *fully connected* en inglés) recibe o vector de características e clasifica a imaxe en función das características extraídas. Na Figura 2.1 amósase un esquema dunha arquitectura típica dunha CNN para un problema de clasificación con 3 clases.

Outro enfoque abordado nos últimos anos para problemas de clasificación de imaxes MSI en teledetección son as GAN. Mediante este tipo de modelos búscase xerar imaxes sintéticas para aumentar os conxuntos de datos de adestramento,

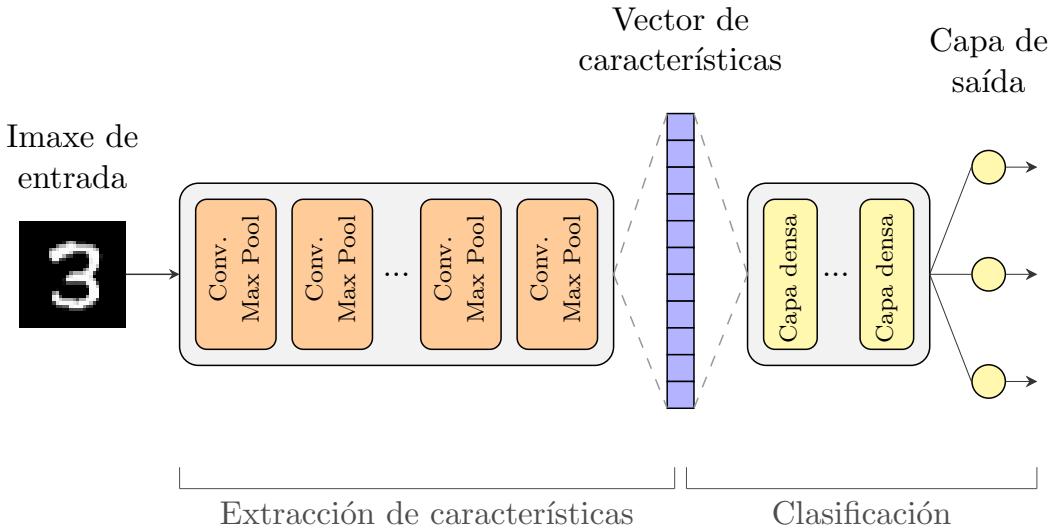


Figura 2.1: Esquema dunha arquitectura típica dunha CNN para un problema de clasificación con 3 clases. A rede comeza cunha serie de capas convolucionais que extraen características das imaxes. A continuación, unha capa densa clasifica as imaxes en función das características extraídas.

para solucionar problemas de desequilibrio de clases e escaseza de datos. Unha vez xeradas as mostras sintéticas, pódese aplicar outro clasificador, como unha CNN, para clasificar as imaxes ou utilizar o discriminador da GAN como clasificador.

Polo xeral, o discriminador dunha GAN consiste nunha CNN que clasifica as imaxes en reais ou sintéticas. Por outro lado, o xerador consiste nunha rede xerativa que produce imaxes sintéticas similares ás do conxunto de adestramento. Para isto, o xerador parte dun vector de ruído e aplica unha serie de capas convolucionais traspostas ata convertir o vector inicial nun tensor que representa unha imaxe sintética, coa mesma resolución e número de canles que as imaxes reais.

O adestramento dunha GAN pasa por adestrar o discriminador e o xerador ao mesmo ritmo. Para isto, en cada época do adestramento, o discriminador recibe unha mostra de imaxes reais e unha mostra de imaxes sintéticas xeradas polo xerador. O obxectivo do discriminador consiste distinguir as imaxes reais das sintéticas. Por outro lado, o xerador busca producir imaxes sintéticas que engañen ao discriminador. Polo tanto, o obxectivo do xerador é producir imaxes sintéticas que o discriminador clasifique como reais.

Dieste et al. [6] propuxeron unha arquitectura de GAN para clasificar imaxes MSI denominada ResBaGAN. Esta rede está formada por un *autoencoder* e unha cGAN. O propósito do autoencoder é aprender a representar as imaxes nun espazo

de características, mentres que a cGAN ten como obxectivo clasificar as imaxes.

O autoencoder está formado por un codificador e un decodificador. A tarefa do codificador consiste en reducir a dimensión da imaxe de entrada a un vector de características, mentres que o decodificador debe reconstruír a imaxe orixinal a partir do vector de características. Na Figura 2.2 amósase un esquema do funcionamento dun autoencoder.

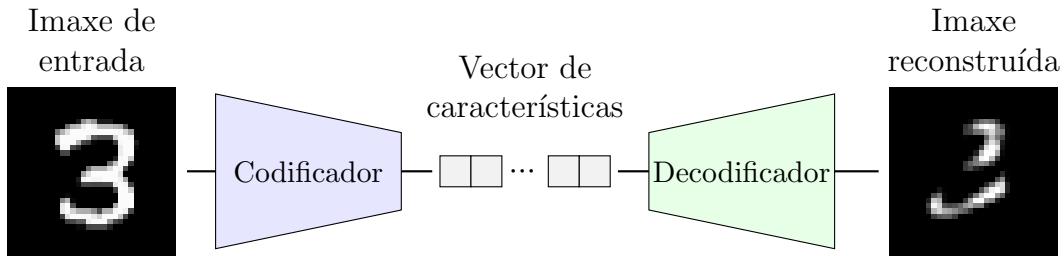


Figura 2.2: Esquema dun autoencoder. O codificador reduce a dimensión da imaxe de entrada a un vector de características, mentres que o decodificador reconstrúe a imaxe a partir do vector de características.

No proceso de adestramento da ResBaGAN, adéstrase primeiro o autoencoder e, a continuación, a cGAN. Desta maneira, distínguense dúas etapas:

Adestramento do autoencoder. O codificador do autoencoder consiste nunha serie de capas convolucionais que reducen a dimensión da imaxe de entrada a un vector de características. O decodificador, pola súa banda, está formado por unha serie de capas convolucionais traspostas que reconstrúen a imaxe a partir do vector de características. Desta maneira, o obxectivo ao adestrar o autoencoder é axustar o codificador e o decodificador para construír un espazo de características que permita representar as imaxes de xeito eficiente.

Adestramento da cGAN. Unha vez adestrado o autoencoder, transfírense os parámetros do codificador ao discriminador da cGAN, e do decodificador ao xerador da cGAN. A continuación, iníciase o adestramento da cGAN para clasificar as imaxes.

O obxectivo desta arquitectura é lograr que o autoencoder aprenda a distribución dos datos de entrada sen considerar as etiquetas de clase. Isto permitirá transferir ese coñecemento á cGAN, de modo que esta comece desde un punto máis estable en lugar de inicializarse de maneira aleatoria.

Os resultados acadados por Dieste et al. [6] demostraron que a arquitectura ResBaGAN é capaz de clasificar imaxes MSI con mellores resultados que outros

modelos de DL, entre os que se inclúen as CNN.

Comparando a arquitectura da ResBaGAN coa StyleGAN2 condicionada, a principal diferenza consiste en que a StyleGAN2 non utiliza un autoencoder para aprender a representar as imaxes nun espazo de características. En cambio, a forma na que a StyleGAN2 condicionada controla as características das imaxes xeradas a través dos “estilos” difire da forma en que a ResBaGAN aprende a representar as imaxes.

A principal contribución deste traballo consiste en adaptar a StyleGAN2 condicionada ao problema de clasificación das imaxes MSI dos 8 ríos galegos, así como a busca da mellor configuración dos hiperparámetros da rede para obter os meillores resultados posibles. Finalmente, compararase a StyleGAN2 condicionada cos resultados acadados pola ResBaGAN co fin de analizar a súa eficacia e rendemento.

Capítulo 3

Metodoloxía

Neste capítulo vaise comentar o preprocessamento que se leva a cabo sobre os imaxes antes de realizar a clasificación destas, así como a arquitectura do modelo empregado e as distintas modificacións que se realizaron para adaptalo ás necesidades do noso traballo.

3.1. Preprocesamento dos datos: segmentación

Inicialmente, o conxunto de datos consiste nunha imaxe MSI da cunca dun río onde cada píxel ten asociado unha etiqueta que indica a clase á que pertence. O obxectivo consiste en adestrar un modelo de clasificación que sexa capaz de clasificar os píxeles da imaxe en función da súa clase. Non obstante, para iso débese preprocessar a imaxe para obter imaxes que poidan ser empregadas como entradas do modelo. O preprocessamento da imaxe pódese dividir en dúas fases: segmentación en superpíxeles e extracción de parches.

A segmentación en superpíxeles é unha técnica de procesamento de imaxes que consiste en dividir unha imaxe en superpíxeles. Un superpíxel é un grupo de píxeles contiguos que comparten características similares, por exemplo, cor, intensidade ou textura. Os superpíxeles non teñen por que ter unha forma concreta nin teñen por que ser regulares.

Existen diferentes algoritmos para realizar a segmentación en superpíxeles [23], como *Simple Linear Iterative Clustering* (SLIC) [1] ou *WaterPixels* (WP) [14]. Cada algoritmo ten as súas propias características e parámetros que poden ser axustados para obter segmentacións con superpíxeles más ou menos grandes, más ou menos regulares, etc. Na Figura 3.1 pode verse como actúa un algoritmo de segmentación en superpíxeles sobre unha imaxe.

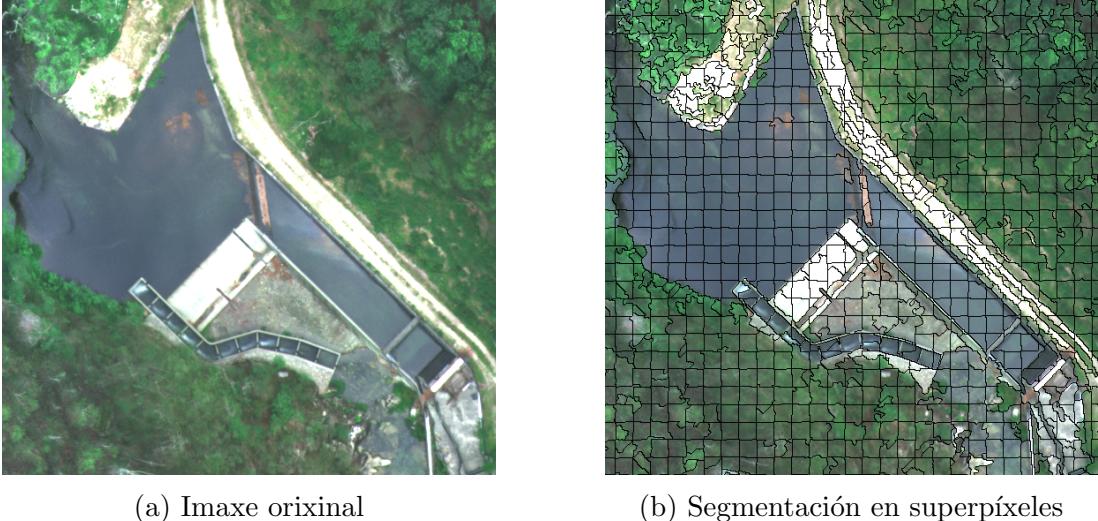


Figura 3.1: Segmentación dunha imaxe en superpíxeles. Obsérvase como os superpíxeles agrupan píxeles con características similares.

A calidade da segmentación depende, principalmente, do problema específico ao que se aplique e dos valores dos parámetros de configuración do algoritmo de segmentación. Por exemplo, a segmentación da Figura 3.1 produce superpíxeles regulares e de tamaño semellante. No ámbito da clasificación MSI, é crucial que a segmentación teña estas características, xa que as imaxes que o modelo utilizará como entrada serán parches de 32×32 píxeles centrados nos centros dos superpíxeles. Deste xeito cada parche é representativo dun superpíxel ou rexión homoxénea da imaxe.

Os parches extráense de xeito que o seu centro coincide co centro do superpíxel. Para calcular o centro do superpíxel, determinase o cadrilátero mínimo que o contén. Así, o punto central deste cadrilátero mínimo considérase o centro do superpíxel. A clase do píxel central do parche determinará a clase do parche. Na Figura 3.2 mósstrase un esquema do proceso de extracción de parches a partir dunha imaxe.

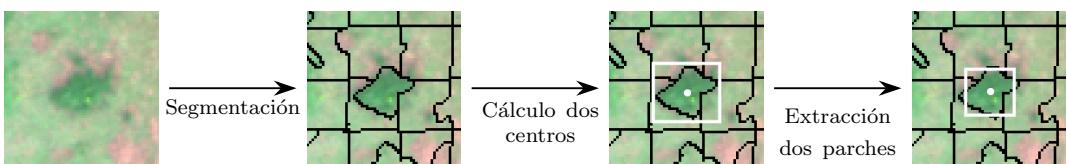


Figura 3.2: Extracción de parches a partir dunha imaxe MSI. En primeiro lugar realiza a segmentación en superpíxeles. A continuación, calcúlanse os centros dos superpíxeles calculando o cuadrilátero mínimo que os contén. Por último extráese un parche de 32×32 píxeles centrado no centro do superpíxel.

3.2. Modelo de clasificación

Como modelo de clasificación empregarase unha cGAN. En particular, utilizarase unha implementación versión condicionada da StyleGAN2 [13] disponible en [5]. De aquí en diante, referirémonos a esta rede como cStyleGAN2. Esta, a pesar de estar codicionada non permite utilizar o discriminador como clasificador. Polo tanto é necesario modificar a saída do discriminador de modo que proporcione unha etiqueta de clase para cada imaxe xerada.

A cStyleGAN2 permite xerar imaxes condicionadas a un vector de clase. Polo tanto, tanto o xerador como o discriminador son capaces de procesar a información de clase. Porén, o discriminador non proporciona unha etiqueta de clase para cada imaxe xerada. Co obxectivo de entender como adaptar o discriminador para que poida ser utilizado como clasificador, primeiro, explicarase brevemente a arquitectura tanto da StyleGAN2, como da súa predecesora, a StyleGAN. A continuación, explicaranse os cambios que levaron a cabo os autores da cStyleGAN2 e darase unha intuición xeométrica sobre como funciona o discriminador e a función de perda asociada a este último. Por último, explicarase como se adaptou o discriminador para que poida ser empregado como clasificador.

3.2.1. Xerador da StyleGAN

Tipicamente o xerador dunha rede xerativa antagónica recibe como entrada un vector de ruído xerado a partir dunha distribución normal ou uniforme. Este vector de ruído pásase por unha capa densa e, posteriormente, trasfórmase nunha imaxe a través dunha serie de bloques van reducindo as canles da imaxe e aumentando a resolución. Na parte esquerda da Figura 3.3 amósase un esquema dun xerador dunha GAN convencional.

Desta maneira, o vector de ruído convértese nun tensor ao pasar polo primeiro bloque e o resto dos bloques van modificando as dimensíons deste tensor ata que se obtén un tensor que representa a imaxe xerada.

Polo xeral, os bloques do xerador están formados por unha capa que aumenta a resolución do tensor (escala o ancho e o alto do tensor) e dúas capas convolucionais que reducen o número de capas. Despois de cada capa, aplícase unha función de normalización. Unha función típica de normalización é a *PixelNorm* definida como:

$$\text{PixelNorm}(x_k) = \frac{x_k}{\sqrt{\frac{1}{N} \sum_{k=1}^c x_k^2 + \epsilon}}, \quad k = 1, \dots, c,$$

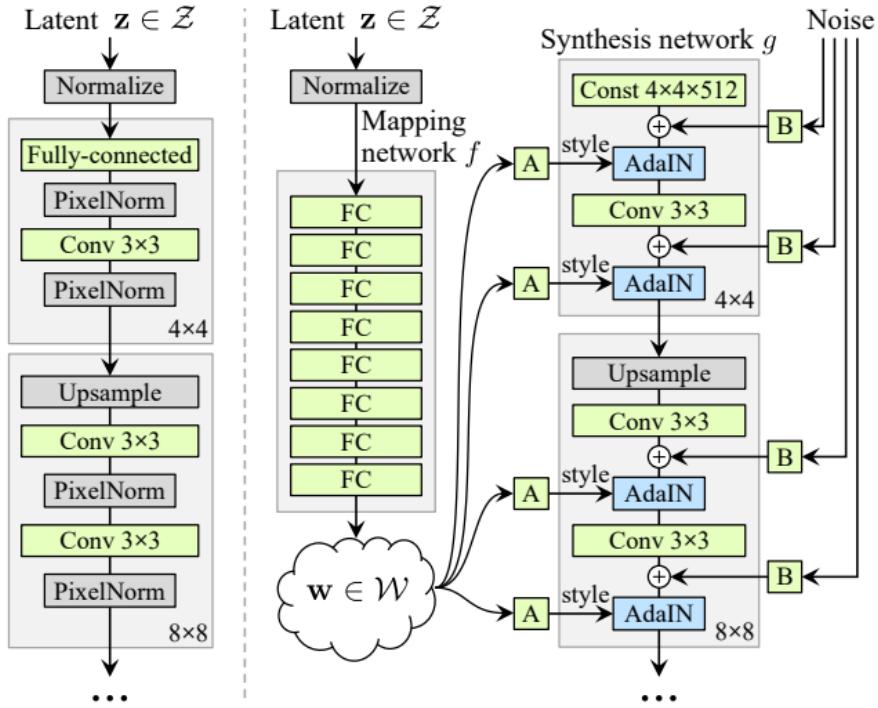


Figura 3.3: Arquitectura dun xerador dunha GAN convencional (esquerda) e dun xerador da StyleGAN (dereita) [9].

onde

- x_k é o valor dun elemento do tensor de entrada (se entendemos o tensor como unha imaxe, a suma dos elementos de $\mathbf{x} = \{x_1, \dots, x_c\}$, $\sum_{k=1}^c x_k$ sería o valor dun píxel da imaxe).
- c é o número de canles do tensor.
- h e w son a altura e a anchura do tensor.
- ϵ é un valor moi pequeno para evitar a división por cero.

O xerador da StyleGAN introduce tres modificacións con respecto ao xerador dunha GAN convencional:

Espazo latente intermedio. Inicialmente, xérase un vector de ruído, $\mathbf{z} \in \mathcal{Z}$, mediante unha distribución normal ou uniforme. O espazo \mathcal{Z} denominase espazo latente. O vector \mathbf{z} transfórmase nun vector $\mathbf{w} \in \mathcal{W}$ a través de varias capas densas. O espazo \mathcal{W} denominase espazo latente intermedio. A continuación, se denotamos por n o número de bloques do xerador, calcúlanse n transformacións afíns (mediante n capas densas cuxos parámetros son

axustados durante o adestramento do xerador) dando lugar aos “estilos”. Os estilos introducíense no xerador a través da normalización AdaIN.

Normalización AdaIN. Para normalizar os valores dos tensores de cada capa, a StyleGAN elimina a normalización *PixelNorm* e introduce unha normalización denominada AdaIN (*Adaptive Instance Normalization*). Esta normalización emprega un vector de estilos para normalizar os valores dos tensores en cada bloque despois de cada capa convolucional. A normalización AdaIN defíñese como:

$$\text{AdaIN}(x_k, \mathbf{s}_k) = s_{s,k} \left(\frac{x_k - \mu(x_k)}{\sigma(x_k)} \right) + s_{b,k},$$

onde

- x_k é o valor dun píxel do tensor de entrada.
- $\mathbf{s}_k = (s_{s,k}, s_{b,k})^T$ é o vector de estilos.
- $\mu(x_k) = \frac{1}{c} \sum_{k=1}^c x_k$, sendo c o número de canles do tensor.
- $\sigma(x_k) = \sqrt{\frac{1}{c} \sum_{k=1}^c (x_k - \mu(x_k))^2}$.

Deste xeito, o vector de estilos de cada bloque ten 2 compoñentes escalares, $s_{s,k}$ e $s_{b,k}$. A compoñente $s_{s,k}$ escala os valores dos tensores e a compoñente $s_{b,k}$, introduce un sesgo. O vector de estilos introducírese no xerador despois de cada capa convolucional mediante a normalización AdaIN.

Ruído engadido. Despois de cada convolución en cada bloque engádese un tensor de ruído xerado mediante unha distribución normal. O tensor de ruído escálase mediante un pesos que tamén se axustan durante o adestramento.

Cabe destacar que o xerador da StyleGAN non parte dun vector de ruído \mathbf{z} , senón dun tensor constante que se axusta durante o adestramento. O vector de ruído transfórmase para acabar xerando n estilos, que se empregan para normalizar os tensores despois de cada convolución en cada bloque.

Na Figura 3.3 pódese ver un esquema do xerador dunha GAN convencional (esquerda) comparado co esquema do xerador da StyleGAN (dereita).

Os autores de [9] xustifican que estas modificacións na arquitectura do xerador da StyleGAN facilitan un control intuitivo e específico dos distintos aspectos da imaxe xerada, mellorando significativamente a calidade das imaxes, a capacidade

de interpolación entre diferentes imaxes e a separación dos factores latentes que inflúen na variabilidade das imaxes.

3.2.2. Xerador da StyleGAN2

A StyleGAN2 introduce principalmente dous cambios con respecto á StyleGAN:

Normalización. Elimínase a normalización AdaIN dos bloques do xerador da StyleGAN. En cambio, traslándose a operación de escalado e sesgo dos valores dos tensores aos pesos das convolucións aplicadas en cada bloque. Polo tanto, na StyleGAN2, a información que aportan os estilos introdúcense nos pesos das capas convolucionais.

Aprendizaxe progresiva. O xerador vai xerando imaxes a partir da saída de cada bloque cuxa resolución vai aumentando. O obxectivo é conseguir que o xerador aprenda a xerar a estrutura da imaxe nos primeiros bloques e os detalles nos últimos. Para isto, a imaxe xerada a partir do bloque i combínase coa imaxe xerada a partir do bloque $i - 1$ mediante unha suma.

Na Figura 3.4 amósase un esquema de como o xerador produce unha imaxe de 32×32 píxeles. O xerador parte dun tensor constante que se axusta durante o adestramento. Este tensor vaise transformando nunha imaxe de 32×32 píxeles a través dos bloques do xerador. O número de bloques depende do tamaño da imaxe final. Para un tamaño de imaxe de 32×32 píxeles, o xerador terá $\log_2(32) - 1 = 4$ bloques.

Os bloques do xerador realizan unha operación de aumento da resolución do tensor (menos o primeiro bloque, que non modifica a resolución), e díusas convolucións. Os estilos introdúcense no proceso modificando os pesos das convolucións. Os tensores de saída de cada bloque transfórmanse en imaxes con 1, 3, 4 ou 5 canles mediante unha convolución que reduce o número de canais do tensor.

As imaxes xeradas a partir das saídas dos bloques teñen cada vez máis resolución. Ademais, para favorecer a aprendizaxe progresiva do xerador, a imaxe xerada a partir da saída do bloque B_i combínase coa imaxe xerada a partir do bloque B_{i-1} mediante unha suma. Deste xeito, os primeiros bloques do xerador aprenden a xerar a estrutura xeral da imaxe, mentres que os últimos bloques encárganse de xerar os detalles.

3.2.3. Discriminador dunha GAN típica

Nesta sección explicarase a arquitectura do discriminador da StyleGAN, en lugar do da StyleGAN2. O discriminador StyleGAN consiste nun discriminador dunha GAN convencional, mentres que o discriminador da StyleGAN2 sufriu modifica-

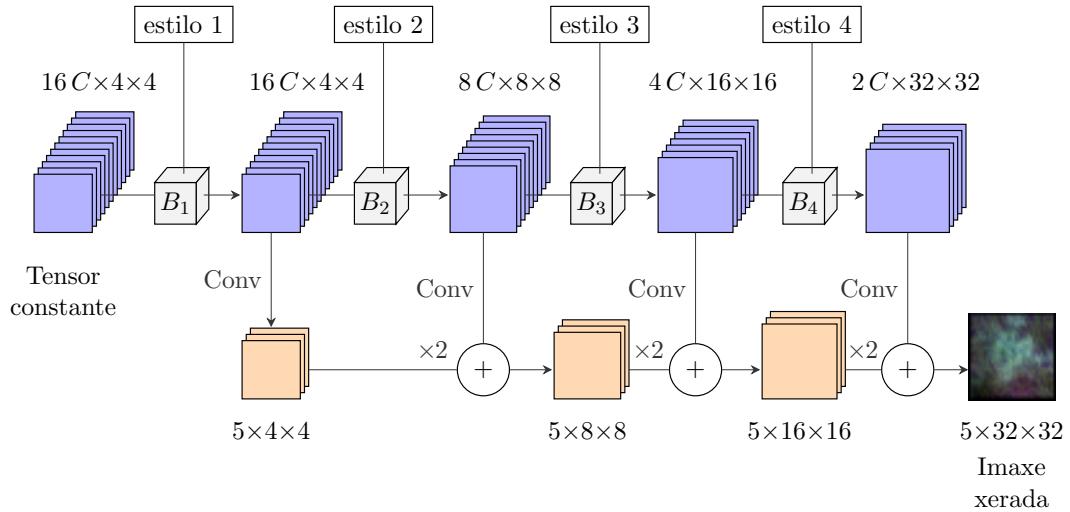


Figura 3.4: Transformación dun tensor constante nunha imaxe de 32×32 píxeles a través dos bloques do xerador da StyleGAN2. O tensor constante pasa por unha serie de bloques do xerador que aumentan progresivamente a súa resolución e reducen o número de canles. Despois de cada bloco, xérase unha imaxe intermedia da mesma resolución que o tensor mediante unha convolución. Esta imaxe intermedia combínase coa imaxe xerada no bloco anterior (tras realizar unha operación que duplica a súa resolución) mediante unha suma.

cións con respecto ao discriminador da StyleGAN inspiradas na arquitectura das redes residuais [10]. Os autores da implementación da cStyleGAN2 baseáronse no discriminador da StyleGAN. Posto que o obxectivo consiste en comprender como adaptar a cStyleGAN2 para que sexa capaz de clasificar imaxes, simplemente se explicará a arquitectura do discriminador da StyleGAN.

Na Figura 3.5 preséntase un esquema dun discriminador dunha GAN convencional. Este está composto dunha serie de capas convolucionais e dunha última capa densa. O discriminador, igual que o xerador, está formado por bloques que dependen do tamaño da imaxe. Para unha imaxe con 32×32 píxeles, o discriminador terá $\log_2(32) = 5$ bloques máis un último bloco que consiste na capa densa. En cada bloque convolucional realízanse catro convoluciones, das cales unha aumenta as canles do tensor de entrada e outra diminúe a resolución do tensor. No último bloco non se aplica a convolución que reduce a resolución do tensor.

Despois de aplicar os bloques convolucionais, a imaxe convertése nun tensor de dimensión $N = 16\ C \times 2 \times 2$, onde C é un hiperparámetro que define a capacidade da rede. Este tensor pódese interpretar como un vector $\mathbf{x} \in \mathbb{R}^N$, sendo \mathbb{R}^N un **espazo de características** da imaxe. Deste modo, a capacidade da rede é un hiperparámetro que permite controlar o número de características que se extraen da imaxe a través das convoluciones. Unha vez que a imaxe atravesa as capas

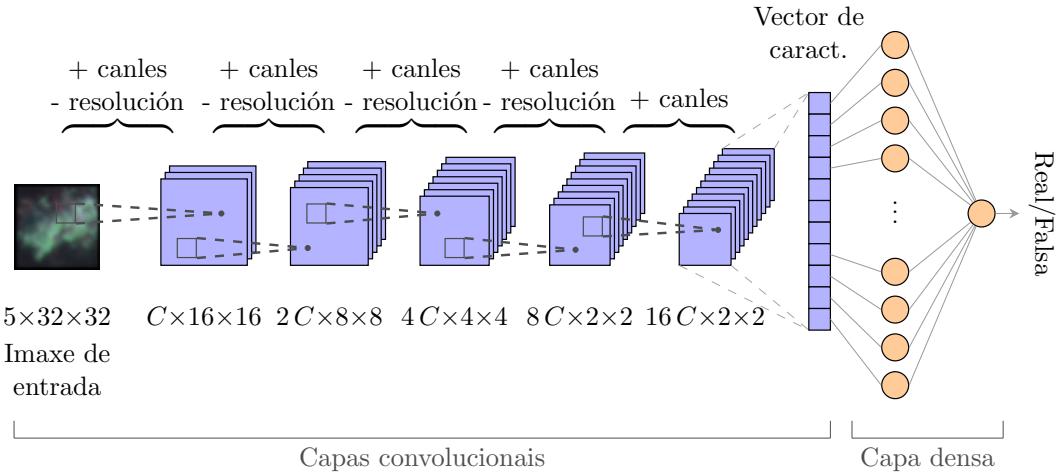


Figura 3.5: As capas convolucionais do discriminador dunha GAN típica transforman unha imaxe nun vector vector de características. Este vector de características é a entrada dunha capa densa que clasifica a imaxe como real ou falsa.

convolucionais, a capa densa do discriminador toma como entrada o vector de características \mathbf{x} e devolve un valor positivo, se considera a imaxe sintética, e negativo, se a considera real.

Os pesos da neurona de saída da capa densa definen un hiperplano en \mathbb{R}^N que separa as clases:

$$\mathbf{p} \cdot \mathbf{x} + b = 0.$$

Na Figura 3.6 amósase unha interpretación gráfica sobre como a capa densa do discriminador separa as imaxes reais e as sintéticas a partir da representación das imaxes no espazo de características. Para poder visualizar o espazo de características supoñamos que $N = 2$, é dicir, que a capa convolucional resume toda a información das imaxes a través de 2 características. Deste modo, a capa densa do discriminador define unha recta que separa as imaxes reais das sintéticas. O vector normal do hiperplano, neste caso da recta, está determinado polos pesos da neurona de saída da capa densa. Deste xeito, se $\mathbf{p} \cdot \mathbf{x} > 0$, o vector \mathbf{x} está no semiplano positivo delimitado polo hiperplano e o discriminador clasificará a imaxe como falsa. Doutro xeito, se $\mathbf{p} \cdot \mathbf{x} < 0$, o vector \mathbf{x} está no semiplano negativo e o discriminador clasificará a imaxe como real.

A función de perda do discriminador consiste na función de perda das SVM, denominada, *hinge loss*:

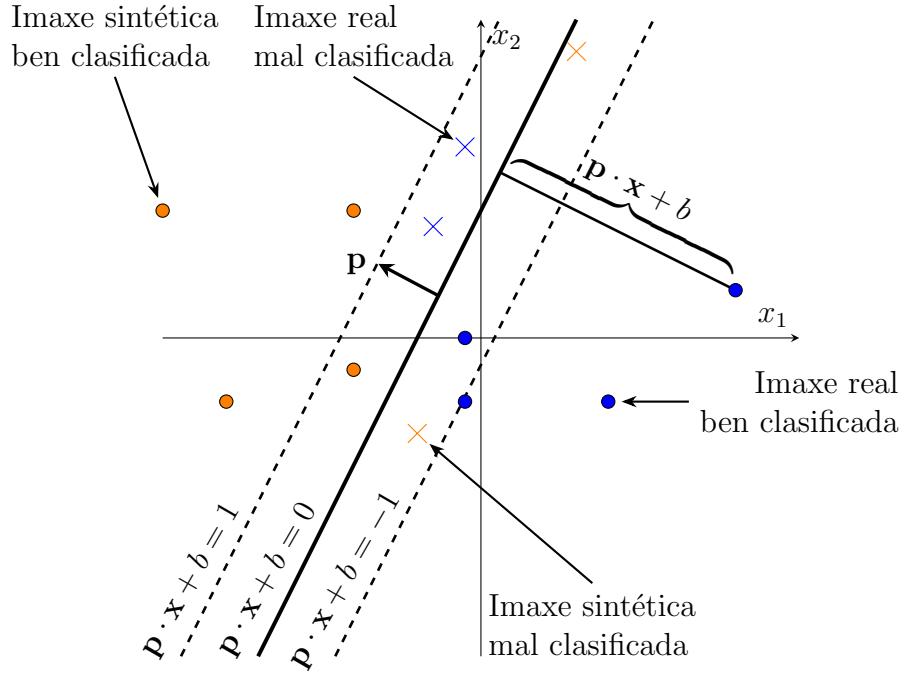


Figura 3.6: Hiperplano de separación das clases no discriminador da StyleGAN. Os parámetros do hiperplano veñen determinados polos pesos da neurona de saída da capa densa. As imaxes represéntanse no espazo a partir do vector de características obtido tras aplicar as capas convolucionais. As imaxes reais represéntanse mediante a cor azul e as imaxes sintéticas mediante a cor vermella. Os errores de clasificación represéntanse mediante unha cruz.

$$\ell(\mathbf{p}, b, \mathbf{x}, y) = \max(0, 1 - y \cdot (\mathbf{p} \cdot \mathbf{x} + b)),$$

onde y é a etiqueta da imaxe (1 para imaxes falsas e -1 para imaxes reais), \mathbf{p} é o vector de pesos da capa densa, b é o sesgo e $\mathbf{x} \in \mathbb{R}^N$ é o vector de características da imaxe obtido tras aplicar as capas convolucionais. O valor $\mathbf{p} \cdot \mathbf{x} + b$ é o valor de saída da capa densa e o valor de saída do discriminador e mide a distancia de \mathbf{x} ao hiperplano de separación. Pódese definir a transformación que se leva a cabo na última capa como:

$$\begin{aligned} d : \mathbb{R}^N &\longrightarrow \mathbb{R} \\ \mathbf{x} &\longmapsto \mathbf{p} \cdot \mathbf{x} + b. \end{aligned}$$

Tendo en conta que o discriminador se avalía cun lote de imaxes reais e sintéticas, a función de perda do discriminador defínese como a suma da perda media para as imaxes reais e da perda media para as imaxes sintéticas:

$$\mathcal{L}_D(\mathbf{p}, b, \mathbf{X}, \mathbf{y}) = \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{R}}^n \ell(\mathbf{p}, b, \mathbf{x}, -1) + \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{F}}^n \ell(\mathbf{p}, b, \mathbf{x}, 1). \quad (3.1)$$

onde \mathcal{R} e \mathcal{F} son os conxunto de imaxes reais e sintéticas coas que se adestra o discriminador en cada lote, respectivamente, e n é o tamaño do lote.

A última capa densa do discriminador e a función de perda do discriminador van xogar un rol importante á hora de condicionar o discriminador para a tarefa de clasificación. Polo tanto, imos analizar con detalle como os valores que toma a función de perda en función de y e $d(\mathbf{x})$.

- Se a imaxe é real ($y = -1$) e o discriminador a clasifica como tal ($d(\mathbf{x}) < 0$), a función de perda resulta

$$\ell(\mathbf{p}, b, \mathbf{x}, -1) = \max(0, 1 - (-1) \cdot d(\mathbf{x})) = \max(0, 1 + d(\mathbf{x})) < 1.$$

No caso de que $d(\mathbf{x}) <= 1$, entón a función de perda é 0. Por exemplo, isto sucede para as imaxes reais que quedan por debaixo do hiperplano definido por $\mathbf{p} \cdot \mathbf{x} + b = -1$ na Figura 3.6.

- Se imaxe é sintética ($y = 1$) e o discriminador a clasifica como tal ($d(\mathbf{x}) > 0$), sucede algo semellante ao caso anterior:

$$\ell(\mathbf{p}, b, \mathbf{x}, 1) = \max(0, 1 - 1 \cdot d(\mathbf{x})) = \max(0, 1 - d(\mathbf{x})) < 1.$$

Os signos de y e $d(\mathbf{x})$ coinciden e o valor da función de perda será menor que 1. Tomará o valor 0 se $d(\mathbf{x}) \leq 1$. De novo, isto sucede para as imaxes sintéticas que quedan por enriba do hiperplano definido por $\mathbf{p} \cdot \mathbf{x} + b = 1$ na Figura 3.6.

- Se a imaxe é real ($y = -1$) e o discriminador a clasifica como sintética ($d(\mathbf{x}) > 0$), os signos de y e $d(\mathbf{x})$ son opostos e a función de perda resulta

$$\ell(\mathbf{p}, b, \mathbf{x}, -1) = \max(0, 1 - (-1) \cdot d(\mathbf{x})) = \max(0, 1 + d(\mathbf{x})) > 1.$$

- A imaxe é sintética ($y = 1$) e o discriminador clasícaa como real ($d(\mathbf{x}) < 0$). Este caso é análogo ao anterior e a función de perda tomará un valor maior que 1.

Cabe destacar que a función de custo penaliza os erros, pero tamén penaliza os acertos cando o discriminador “non está moi seguro”. Estes casos prodúcense cando $d(\mathbf{x}) \in (0, 1)$ e $d(\mathbf{x})$ e y teñen o mesmo signo.

Nas seccións anteriores non se comentou a función de perda do xerador. O motivo reside en que está moi relacionada coa do discriminador e formúlase de forma natural unha vez que se introduce a do discriminador. Posto que o adestramento do xerador consiste en xerar imaxes e tratar de enganar o discriminador, a función de perda do xerador deberá ser maior tanto máis seguro estea o discriminador de que a imaxe xerada polo xerador é sintética. Lembremos que o xerador clasifica as imaxes como sintéticas se $d(\mathbf{x}) > 0$ e que a “seguridade da clasificación” vén determinada por $|d(\mathbf{x})|$. Desta maneira, se n é o tamaño do lote, a función de perda do xerador defínese como a media de $d(\mathbf{x})$, sendo \mathbf{x} o vector de características da imaxe xerada polo xerador:

$$\mathcal{L}_G(\mathbf{w}, b, \mathbf{x}) = \frac{1}{n} \sum_{i=1}^n d(\mathbf{x}_i), \quad (3.2)$$

Por defecto, o xerador da StyleGAN2 utiliza (3.1) e (3.2) como función de perda do discriminador e do xerador, respectivamente. Aínda que a implementación da StyleGAN2 ofrece a posibilidade de empregar outra función de perda, a versión condicionada da StyleGAN2 soamente permite empregar estas funcións de perda.

3.2.4. StyleGAN2 condicionada

A StyleGAN2 condicionada é unha versión da StyleGAN2 que permite xerar imaxes condicionadas a unha etiqueta de clase. Para introducir a información de clase na rede, é necesario modificar o xerador para que utilice a información de clase dalgunha forma para xerar as imaxes. Por outra banda, o discriminador debe adaptarse para que sexa capaz de clasificar as imaxes en reais ou sintéticas condicionado á clase da imaxe.

Lembremos que o xerador da StyleGAN2 converte un vector de ruído $\mathbf{z} \in \mathcal{Z}$ nun vector $\mathbf{w} \in \mathcal{W}$, sendo \mathcal{W} un espazo latente intermedio. Para introducir a información de clase no xerador resérvase unha parte do espazo \mathcal{W} para codificar a información de clase.

Por exemplo, nun problema de clasificación con 10 clases, a clase pódese codificar mediante un vector de 10 compoñentes (*one-hot encoding*) que soamente ten un elemento distinto de cero que representa o índice da clase. Este vector concaténase co vector \mathbf{w} para formar un vector \mathbf{w}' que se emprega para xerar os estilos. Polo tanto, o vector \mathbf{w}' ten dimensión $d + m$, onde d é a dimensión do espazo latente

intermedio e m é o número de clases. Ademais, para dous imaxes da mesma clase, as m últimas componentes do vector \mathbf{w}' serán as mesmas. Polo tanto, a información de clase no xerador introdúcese a través dos estilos.

O discriminador da StyleGAN e da StyleGAN2 soamente ten unha saída, pois só debe determinar se unha entrada é sintética ou real. Para a versión condicionada do xerador, modifícase a última capa para que produza un valor por clase. Para implementar isto, débese modificar a última capa do discriminador para que teña m saídas, unha por cada clase. Na Figura 3.7 preséntase o discriminador da StyleGAN2 condicionada para un problema con 3 clases.

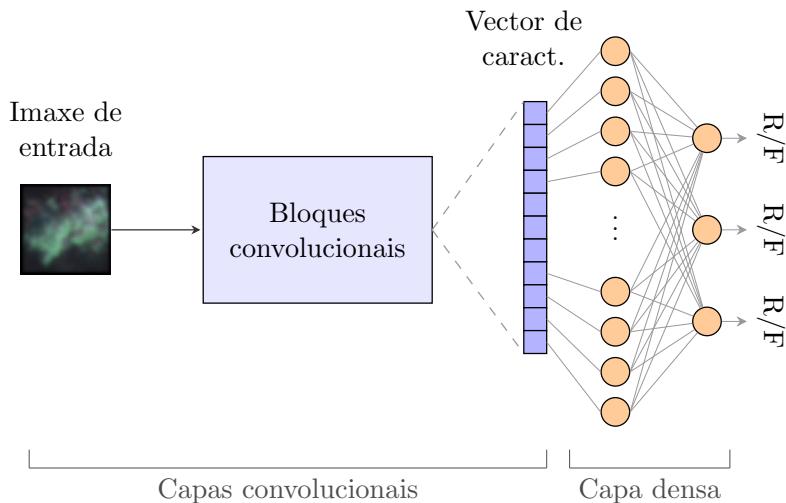


Figura 3.7: O discriminador da StyleGAN2 condicionada engade unha neurona de saída por cada clase do problema de clasificación. Cada unha destas neuronas especialízase en clasificar en reais ou falsas soamente as imaxes dunha das clases do problema.

Segundo a liña da interpretación xeométrica que se deu na Sección 3.2.3, os pesos das neuronas da última capa definen m hiperplanos. Cada hiperplano adéstrase para separar as imaxes reais das sintéticas dunha clase concreta. Para que isto sexa posible, se a imaxe que se introduce como entrada do discriminador pertence á i -ésima clase, a saída do discriminador será a saída da neurona i -ésima. Deste modo, a función de perda do discriminador calcula a perda soamente da clasificación da imaxe en real ou sintética da clase á que pertence.

Vexámolo dun xeito máis formal. Sexan $\mathbf{p} = \{\mathbf{p}^1, \dots, \mathbf{p}^m\}$, $\mathbf{p}^j \in \mathbb{R}^N$ os pesos da última capa do discriminador. As saídas da última capa do discriminador para unha imaxe \mathbf{x} , é un vector $\mathbf{d} = \{d^1(\mathbf{x}), \dots, d^m(\mathbf{x})\}$, onde $d^j(\mathbf{x}) = \mathbf{p}^j \cdot \mathbf{x} + b^j$. Se a imaxe pertence á clase i , a perda do discriminador para a imaxe é:

$$\mathcal{L}_D(\mathbf{p}, b, \mathbf{x}, y) = \ell(\mathbf{p}^i, b^i, \mathbf{x}, y) = \max(0, 1 - y \cdot d^i(\mathbf{x})).$$

onde y toma o valor 1 se a imaxe é real e -1 se a imaxe é sintética. Da mesma maneira que antes, para un lote de imaxes, a función de perda do discriminador defínese como a media da perda para cada imaxe do lote. Durante o adestramento do xerador, a perda do mesmo para unha imaxe xerada \mathbf{x} é $d^i(\mathbf{x})$, onde i é a clase á que pertence a imaxe.

3.2.5. Adaptación da StyleGAN2 condicionada para clasificación

Para que o discriminador poida ser utilizado para clasificar debe ser capaz dalgún modo de determinar a clase da imaxe. Posto que soamente imos utilizar o discriminador para clasificar imaxes reais, o discriminador pode devolver o índice da clase “máis real”. Desta maneira, a clase dunha imaxe real será

$$\hat{y} = \arg \min_i d^i(\mathbf{x}). \quad (3.3)$$

Xeométricamente, para obter a clase dunha imaxe real dado o seu vector de características \mathbf{x} , estase minimizando o produto escalar entre o vector de pesos da última capa do discriminador e o vector de características da imaxe \mathbf{x} más o sesgo:

$$\min_i \mathbf{p}^i \cdot \mathbf{x} + b^i.$$

Noutras palabras, estase buscando o hiperplano que máis deixa a \mathbf{x} no semiplano negativo e que máis lonxe está de \mathbf{x} . No caso de que \mathbf{x} estea no semiplano positivo de todos os hiperplanos, a clase da imaxe será a clase virá determinada polo hiperplano máis próximo a \mathbf{x} . Na Figura 3.8 amósase unha interpretación xeométrica da clasificación dunha imaxe real polo discriminador.

Cabe destacar que no caso de que todos os hiperplanos deixen a \mathbf{x} no semiplano positivo, o valor da función de perda do discriminador será alto, pois a imaxe clasificarse como sintética, e o estarase cometendo un erro de clasificación.

Deste modo, partindo da versión condicionada da StyleGAN2 [5], adaptouse o discriminador para que devolva a clase dunha imaxe de acordo coa Expresión (3.3). A implementación da StyleGAN2 condicionada para clasificación está dispoñible en [2].

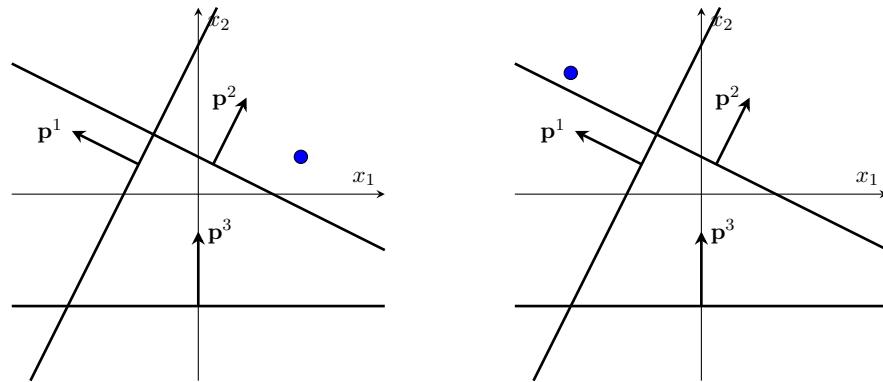


Figura 3.8: Interpretación xeométrica da clasificación do vector de características \mathbf{x} (representado mediante un punto de cor azul) dunha imaxe real polo discriminador. Na esquerda, asignaráselle a \mathbf{x} a clase 1, pois o hiperplano con vector normal \mathbf{p}^1 é o que máis lonxe está de \mathbf{x} e que deixa a \mathbf{x} no semiplano negativo. Na dereita, todos os hiperplanos deixan a \mathbf{x} no semiplano positivo, polo que a clase de \mathbf{x} será a clase do hiperplano más próximo a \mathbf{x} , neste caso, a clase 2.

Capítulo 4

Materiais e métricas

Neste capítulo introduciranse os materiais empregados para a realización das probas e as métricas de avaliación utilizadas para comparar os resultados obtidos cos de outros traballos.

4.1. Conxuntos de datos

Os conxuntos de datos empregados para avaliar o modelo de clasificación están formados por imaxes multiespectrais tomadas dende un *Vehículo Aéreo non Tripulado* (UAV) das cuncas de 8 ríos galegos. As imaxes foron tomadas co obxectivo de monitorizar automaticamente as cuncas dos ríos de Galicia. Especificamente, preténdese determinar o estado da vexetación, identificar áreas ocupadas por especies invasoras e detectar estruturas artificiais que ocupan a cunca do río empregando imaxes multiespectrais.

En particular, trátase de 8 imaxes de alta resolución multiespectral que foron utilizadas en [3] para avaliar diversos algoritmos de segmentación e de clasificación. As imaxes foron capturadas entre 2018 e 2020 voando un UAV a 120 metros de altitude sobre varias cuncas de ríos en Galicia, resultando nunha resolución espacial de 10 cm/píxel. As imaxes están formadas por cinco bandas correspondentes ás lonxitudes de onda de 475 nm (azul), 560 nm (verde), 668 nm (vermello), 717 nm (*red-edge*) e 840 nm (infravermello próximo). O Cadro 4.1 detalla as localizacións e dimensíons específicas das imaxes.

Todos os conxuntos de datos foron segmentados en superpíxeles empregando o algoritmo WP escollendo un tamaño medio de 400 px/superpíxel, cun mínimo de 100 px/superpíxel, e usando un factor de compacidade de 0.5 puntos, seguindo o enfoque de [3]. Os parches extraídos teñen unha dimensión espacial de 32×32

píxeles. Adicionalmente, todos os datos foron normalizados no rango $[-1, 1]$.

Na Figura 4.1 pódense observar as imaxes en RGB (esquerda) xunto coa súa información de referencia (dereita) para os 8 ríos. O Cadro 4.2 amosa o número de mostras por clase en cada conxunto de datos.

Antes de adestrar o modelo de clasificación coas imaxes MSI, comprobarase o funcionamento e rendemento do modelo de clasificación co conxunto de datos MNIST [17]. Este conxunto de datos consiste nunha colección de imaxes de díxitos manuscritos de 28×28 píxeles, etiquetadas co díxito que representan. Trátase dun conxunto de datos sinxelo, empregado a miúdo para comparar o rendemento de distintos modelos de clasificación de imaxes, polo que resulta ideal para validar a implementación do modelo.

Conxunto de datos	Localización	Tamaño	Píxeles etiquetados	Superpíxeles etiquetados
Río Oitavén	$42^{\circ}22'15.48''$ N $8^{\circ}25'47.07''$ O	6722×6689 (760×760 m 2)	22.04 M	35 838
Cunca do Xesta	$42^{\circ}23'34.95''$ N $8^{\circ}21'21.23''$ O	$17\ 202 \times 3\ 848$ ($1\ 945 \times 435$ m 2)	40.00 M	82 218
Encoro de Eiras	$42^{\circ}20'45.24''$ N $8^{\circ}21'21.23''$ O	$18\ 221 \times 5\ 176$ ($2\ 260 \times 660$ m 2)	38.35 M	27 865
Regato Ermidas	$42^{\circ}22'48.43''$ N $8^{\circ}24'53.36''$ O	$18\ 972 \times 11\ 924$ ($2\ 190 \times 1\ 390$ m 2)	65.56 M	36 063
Río Ferreiras	$43^{\circ}32'46.96''$ N $7^{\circ}57'16.66''$ O	$9\ 219 \times 9\ 335$ (740×750 m 2)	40.19 M	82 856
Río das Mestas	$43^{\circ}38'30.24''$ N $7^{\circ}58'44.08''$ O	$9\ 040 \times 4\ 915$ (920×510 m 2)	27.17 M	57 058
Río Mera	$43^{\circ}34'31.15''$ N $7^{\circ}52'34.81''$ O	$22\ 116 \times 10\ 718$ ($2\ 770 \times 1\ 370$ m 2)	99.24 M	125 142
Río Ulla	$42^{\circ}49'14.32''$ N $7^{\circ}54'5.29''$ O	$16\ 555 \times 4\ 220$ ($1\ 420 \times 380$ m 2)	46.01 M	3 548

Cadro 4.1: Descripcións das imaxes multiespectrais das cuncas dos 8 ríos galegos. O número de superpíxeles etiquetados foi obtido a través de segmentación con WP.

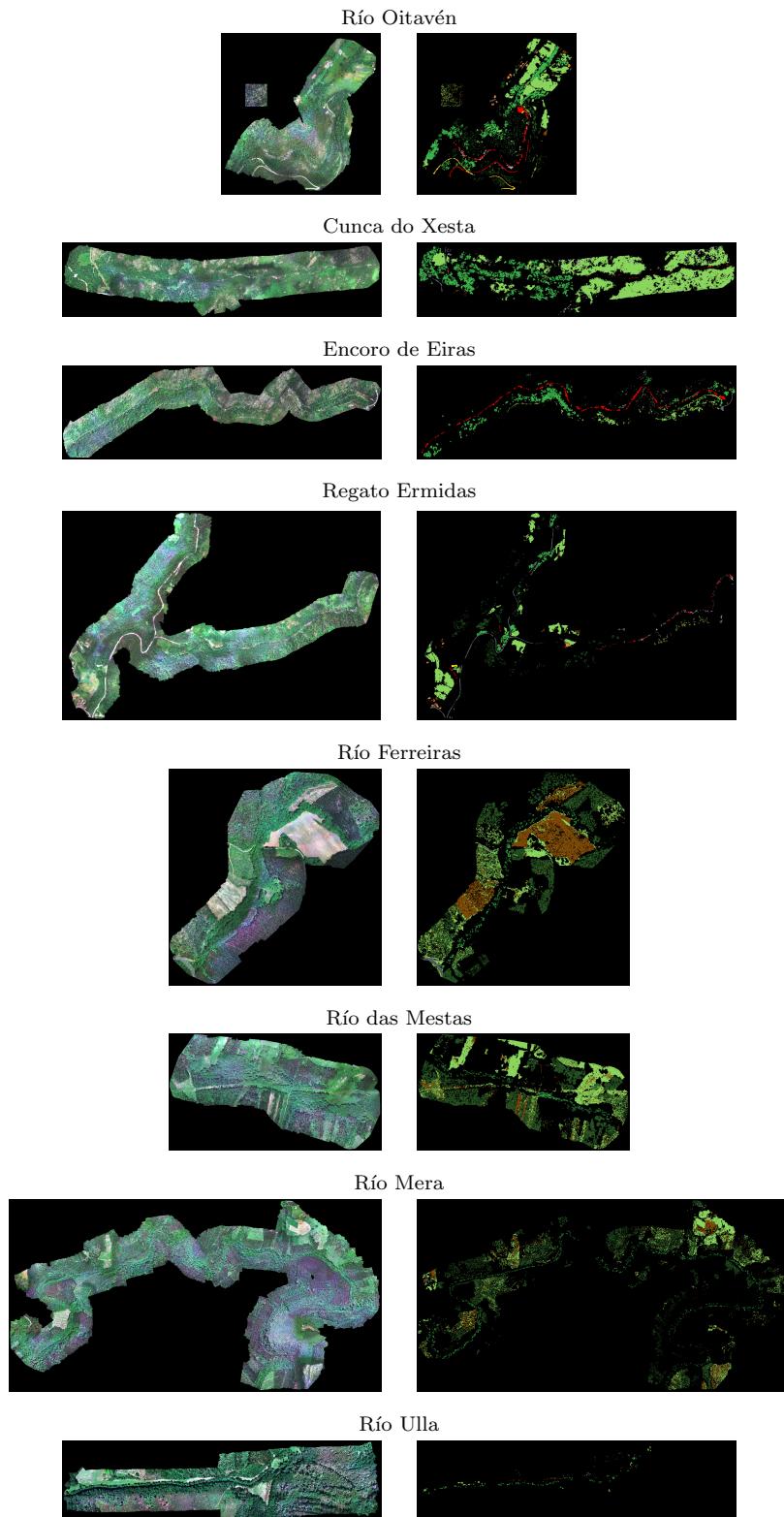


Figura 4.1: Imaxes multiespectrais dos 8 ríos galegos en RGB (esquerda) xunto coa súa información de referencia (dereita). A clase correspondente a cada cor na información de referencia está descrita no Cadro 4.2, mentres que a cor negra significa que non hai información de referencia sobre devanditos píxeles.

#	Cor	Clase	Río Oitavén	Cunca do Xesta	Encoro de Eiras	Regato Ermidas	Río Ferreiras	Río das Mestas	Río Mera	Río Ulla
1		Auga	1 515	488	3 557	890	0	0	0	0
2		Terra	978	411	867	874	24 377	7 076	18 535	495
3		Pedras	749	4 632	2 923	1 846	0	0	0	0
4		Asfalto	202	700	340	3 597	685	0	395	32
5		Formigón	774	0	182	126	0	0	10	0
6		Tellados	448	0	47	608	0	0	18	0
7		Prados	9 376	52 276	5 722	13 052	22 364	23 417	43 191	494
8		Árbores nativas	11 300	23 711	13 466	4 559	4 830	2 973	13 241	2 243
9		Piñeiros	2 109	0	687	1 527	14	0	0	284
10		Eucaliptos	8 387	0	74	8 984	30 586	23 592	49 752	0
Mostras totais:			35 838	82 218	27 865	36 063	82 856	57 058	125 142	3 548

Cadro 4.2: Número de parches por clase en cada conxunto de datos. Os parches obtivéronse a partir da segmentación en superpíxeles con WP.

4.2. Métricas de avaliación

Para avaliar o rendemento do modelo de clasificación empregáronse as seguintes métricas de avaliación:

- OA: porcentaxe total de parches correctamente clasificados.
- AA: media das porcentaxes de parches correctamente clasificados para cada clase.

A *Average Accuracy* (AA), a diferenzia da *Overall Accuracy* (OA), ten en conta a información de clase. Deste modo, ante conxuntos de datos desbalanceados, como é o caso, a OA pode ser engañosa, xa que unha clasificación correcta nunha clase maioritaria pode compensar unha clasificación incorrecta nunha clase minoritaria. Pola contra, a AA é unha métrica que ten en conta todas as clases por igual, polo que é máis robusta fronte a desbalanceos.

No ámbito da teledetección é habitual empregar estas dúas métricas a nivel de píxel, en lugar de a nivel de parche. Polo tanto, para levar a cabo a comparación coa ResBaGAN, utilizaranse as mesmas métricas pero a nivel de píxel, tal e como se describe en [3].

4.3. Entorno de execución

A realización das probas levouse a cabo a través dos servizos do Centro de Supercomputación de Galicia (CESGA) que ofrece mediante conexión *Secure Shell* (SSH). A continuación comentaranse os recursos hardware e software em-

pregados para a realización das probas.

4.3.1. Hardware

As probas leváronse a cabo nos nodos interactivos que pon a disposición o CESGA. A configuración empregada foi a seguinte:

- Modelo da *Unidade de Procesamento Gráfica* (GPU): Tesla T4 con 16 GB de memoria [4].
- Modelo da *Unidade de Procesamento Central* (CPU): Intel(R) Xeon(R) Platinum 8352Y CPU @ 2.20GHz
- Memoria *Memoria de Acceso Aleatorio* (RAM) de 16 GB

4.3.2. Software

Os nodos interactivos do CESGA contan cun sistema operativo Rocky Linux 8.4 (Green Obsidian) e coa versión 11.2.152 de *Arquitectura Unificada de Dispositivos de Cómputo* (CUDA). Para a execución das probas creouse un entorno virtual con Conda (versión 23.7.4) coa versión 3.9.18 de Python. As principais bibliotecas empregadas foron:

- torch 1.2.0+cu92.
- torchvision versión 0.4.0+cu92.

O resto das bibliotecas espeficícanse no ficheiro `environment.yml` que se atopa no repositorio do proxecto en GitHub [2].

Capítulo 5

Probas

Neste capítulo preséntase o plan de probas que se levou a cabo para adestrar e avaliar o modelo de clasificación cos distintos conxuntos de datos presentados no capítulo anterior. Ademais, tamén se describen os experimentos realizados e os resultados obtidos.

5.1. Plan de probas

Nesta sección coméntanse os experimentos levados a cabo. Cada experimento está formado por unha serie de adestramentos e avaliacións do modelo de clasificación. Para cada adestramento especifícanse os hiperparámetros do modelo e o conxunto de datos empregado. Os experimentos realizados foron os seguintes:

Validación do modelo para xerar mostras sintéticas. En primeiro lugar, validouse a implementación da funcionalidade relativa á xeración de imaxes sintéticas condicionadas a unha etiqueta de clase do modelo e a capacidade de clasificación deste. Para isto, empregouse o conxunto de datos MNIST.

Optimización de hiperparámetros. Unha vez validada a implementación, levouse a cabo unha optimización de hiperparámetros para atopar os valores óptimos. Para isto, empregouse o conxunto de datos do río Oitavén.

Comparación con outros modelos de clasificación. Finalmente, avaliouuse o modelo de clasificación con todos os conxuntos de datos multiespectrais presentados no capítulo anterior coa configuración de hiperparámetros que proporcionou un mellor resultado co conxunto de datos do río Oitavén. O obxectivo é comprobar se os conxuntos de datos xerados melloran a clasificación con respecto a cando se aprende únicamente de datos reais

e comparar os resultados acadados cos obtidos pola ResBaGAN.

A optimización de hiperparámetros dividerémola en 3 subexperimentos, un por cada hiperparámetro ou grupo de hiperparámetros axustados.

Co obxectivo de presentar os resultados de forma clara, no Cadro 5.1 descríbese a numeración dos adestramentos que forman parte de cada experimento.

Experimentos	Subexperimentos	Adestramentos
Experimento 1: Validación		1-4
Experimento 2: Optimización	LR_G e LR_D	5-20
	Capacidade da rede	21-28
	LR adaptativo	29-37
Experimento3: Comparación		38-44

Cadro 5.1: Esquema dos esperimentos a realizar e numeración dos adestramentos que forman parte dos experimentos.

5.1.1. Experimento 1: Validación do modelo de clasificación

En primeiro lugar, levouse a cabo un experimento para validar a implementación do modelo de clasificación. En concreto, o obxectivo consiste en comprobar se as imaxes sintéticas xeradas polo modelo se parecen ás orixinais do conxunto de datos e se o modelo obtén un bo rendemento en canto á clasificación utilizando as métricas de avaliación presentadas na Sección 4.2. Para isto, empregouse o conxunto de datos MNIST.

En particular, realizáronse 4 adestramentos nos que se probou con distintos valores para a *Taxa de Aprendizaxe* (LR) do xerador e do discriminador. Ao valor por defecto da LR de ambos, $2 \cdot 10^{-4}$, engadiuse un valor lixeiramente inferior, $1 \cdot 10^{-4}$ co fin de observar se se obtén un mellor rendemento. No Cadro 5.2 indícanse os valores da LR empregados para cada adestramento.

O obxectivo deste experimento non é acadar os melhores resultados posibles, senón simplemente comprobar a capacidade do modelo para xerar imaxes sintéticas e clasificalas. Debido a isto, non se realizou unha optimización de hiperparámetros extensa.

	Adest. 1	Adest. 2	Adest. 3	Adest. 4
LR _G	$2 \cdot 10^{-4}$	$2 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$
LR _D	$2 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$2 \cdot 10^{-4}$	$1 \cdot 10^{-4}$

Cadro 5.2: Valores da LR do xerador e do discriminador nos adestramentos co conxunto de datos MNIST.

Utilizouse unha división en tres subconxuntos do conxunto de datos MNIST: adestramento, validación e test. O conxunto de test ten 10 000 mostras, o de adestramento, 59 000, e o de validación 1 000. Para que as imaxes sexan compatibles co modelo, redimensionáronse a un tamaño de 32×32 píxeles engadindo un marxe de 2 píxeles en cada lado.

5.1.2. Preprocesamento das imaxes multiespectrais

Para os seguintes experimentos utilizáronse os conxuntos de datos multiespectrais obtidos a partir das imaxes multiespectrais de observación terrestre correspondentes a ríos galegos. Os parches extraéreronse mediante unha segmentación en superpíxeles utilizando o algoritmo WP cos parámetros especificados na Sección 4.1. De aquí en diante, referirémonos ao conxunto dos parches extraídos como o conxunto de datos da imaxe multiespectral correspondente.

Con respecto aos adestramentos do experimento 2 (5-37), utilizouse unha división do conxunto de datos en dous subconxuntos: adestramento e validación. No Cadro 5.3 amósase o número de mostras de cada clase que forman parte de ambos conxuntos. Non se empregou un conxunto de test, debido a que o número de mostras dalgunhas clases é moi limitado. Desta maneira, para comparar os resultados obtidos cos distintos hiperparámetros, utilizouse o conxunto de validación.

O feito de empregar poucas mostras para o conxunto de adestramento débese a dous motivos: lograr un equilibrio entre as clases e conseguir un rendemento bo do modelo con poucas mostras.

Para os adestramentos do terceiro experimento (38-44), si que se utilizou unha división en tres subconxuntos. Esta será a mesma que a utilizada por Dieste et al. para adestrar a ResBaGAN [6]. Deste modo, o conxunto de adestramento conta co 15 % das mostras de cada clase e o de validación, co 5 %. Posto que a avaliación do modelo se realiza a nivel de píxel, en lugar de a nivel de segmento, o conxunto de test contén todos os píxeles etiquetados da imaxe, excepto os píxeles que se usaron para validación e adestramento, asegurando que os datos de test sexan independentes.

Clase	Adestramento	Validación
Auga	200	200
Terra	200	200
Pedras	200	149
Asfalto	64	64
Formigón	200	200
Tellados	200	94
Prados	200	200
Árbores Nativas	200	200
Piñeiros	200	200
Eucaliptos	200	200

Cadro 5.3: Número de mostras de adestramento e validación para cada unha das clases presentes na imaxe do Río Oitavén para os adestramentos do experimento 2 (5-37).

A continuación, detállase como se realiza a avaliación da clasificación co conxunto de test a nivel de píxel. En primeiro lugar, obtéñense as predicións do modelo para os parches do conxunto de test, é dicir, todos os parches do conxunto de datos da imaxe multiespectral correspondente. A continuación, esténdese a clasificación realizada sobre cada parche a todos os píxeles do segmento cuxo centro coincide co parche. Desta maneira, constrúese un mapa de clasificación que involucra todos os píxeles da imaxe. A continuación, elimínanse do mapa de clasificación os píxeles do centro dos segmentos utilizados para adestramento e validación. Finalmente, compárase o mapa de clasificación resultante co mapa de referencia e calcúlanse as métricas que correspondan, como poden ser a OA e a AA. Desta maneira, exclúense soamente os píxeles do centro dos segmentos utilizados para adestrar e validar o modelo.

5.1.3. Experimento 2: Optimización de hiperparámetros

Para comprobar o rendemento do modelo con distintos hiperparámetros empréga-se o conxunto de datos do río Oitavén debido a que é un dos que contén mostras de todas as clases e o que está máis balanceado. O obxectivo é atopar unha combinación de hiperparámetros que produza bons resultados en canto á clasificación para utilizar no experimento 3 co resto dos conxuntos de datos multiespectrais.

A optimización de hiperparámetros subdivídese en outros 3 experimentos:

- Axuste da LR do xerador e do discriminador.

- Axuste da capacidade da rede.
- Introdución dunha *Taxa de Aprendizaxe Adaptativa* (LR-ADA) para o xerador e o discriminador.

Axuste da LR do xerador e do discriminador

Probáronse distintas combinacións de valores para a LR do xerador e do discriminador. En particular, leváronse a cabo 16 adestramentos froito das combinacións das seguintes LR para o discriminador e para o xerador:

$$\begin{aligned} \text{LR}_G &\in \{2 \cdot 10^{-4}, 1 \cdot 10^{-4}, 5 \cdot 10^{-5}, 2,5 \cdot 10^{-5}\}. \\ \text{LR}_D &\in \{2 \cdot 10^{-4}, 1 \cdot 10^{-4}, 5 \cdot 10^{-5}, 2,5 \cdot 10^{-5}\}. \end{aligned}$$

Nas primeiras filas do Cadro 5.6 numéransen os adestramentos que forman parte deste experimento do 5 ao 20. Cada adestramento correspón dese con unha combinación de valores para a LR do xerador e do discriminador.

Axuste da capacidade da rede

A capacidade da rede é un hiperparámetro que afecta tanto ao discriminador como ao xerador. Con respecto ao discriminador, os filtros das convolucións dependen da capacidade da rede. Deste xeito, como se reflexa na Figura 3.5, segundo se aumenta a capacidade da rede, o número de canles dos tensores producidos en cada saída dos bloques do discriminador aumenta. Como consecuencia, o espazo de características obtido como saída dos bloques convolucionais será de maior tamaño. Noutras palabras, ao aumentar a capacidade da rede, auméntanse as características aprendidas polo discriminador para despois levar a cabo a clasificación.

De forma semellante ao discriminador, no caso do xerador a capacidade da rede inflúe no número de canles do tensor de partida para xerar as imaxes e no tamaño dos filtros das convolucións. Na Figura 3.4 pódese ver como os tensores de saída de cada bloque teñen más ou menos canles segundo a capacidade da rede.

Así, fixando un valor para a LR do discriminador, probáronse distintas capacidades da rede variando a LR do xerador. En particular, leváronse a cabo 8 adestramentos, froito das combinacións dos seguintes valores para a LR do xerador e para a capacidade da rede:

$$\begin{aligned} \text{LR}_G &\in \{2 \cdot 10^{-4}, 1 \cdot 10^{-4}, 5 \cdot 10^{-5}, 2,5 \cdot 10^{-5}\}. \\ C &\in \{8, 32\}. \end{aligned}$$

Cabe mencionar que o valor por defecto para a capacidade da rede é 16, polo que no experimento descrito na Sección 5.1.3 xa se levaron a cabo adestramentos con estes valores para a capacidade da rede.

No Cadro 5.6 numéranse do 21 ao 28 os adestramentos que se corresponden coas combinacións de hiperparámetros empregadas neste subexperimento.

Introdución dunha LR-ADA para o xerador e o discriminador

Co obxectivo de evitar o sobreaxuste do discriminador, introduciuse unha LR-ADA para o xerador e o discriminador. O método empregado para reducir a LR consiste en multiplicar a LR por un certo valor γ en certas épocas do adestramento. Para isto, utilizarase a clase *MultiStepLR* da biblioteca *PyTorch* [18].

Para determininar as épocas nas que se diminuirá a LR, determinouse unha época na que comezar a reducir a LR (*epoca_inicial*), unha época na que se deixou de reducir (*epoca_final*), e un paso (*paso_epoca*) co que obter as épocas nas que se reducirá a LR a partir de *epoca_inicial*. Deste modo, especificando un valor inicial (LR_i) e final (LR_f) para a LR, se n determina o número de épocas nas que se reducirá a LR, pódese calcular γ como:

$$\gamma = \left(\frac{LR_f}{LR_i} \right)^{1/n}.$$

Utilizando este método para diminuír a LR durante o adestramento, probáronse tres formas de diminuír a LR do xerador e do discriminador:

1. Redución soamente da LR do discriminador:
 - Xerador: $LR_{G,f} = LR_{G,i}$.
 - Discriminador: $LR_{D,f} = \frac{1}{4}LR_{D,i}$.
2. Redución da LR do xerador e do discriminador:
 - Xerador: $LR_{G,f} = \frac{3}{4}LR_{G,i}$.
 - Discriminador: $LR_{D,f} = \frac{1}{8}LR_{D,i}$.
3. Redución da LR do xerador e do discriminador, reducindo máis que no caso anterior a LR do xerador:
 - Xerador: $LR_{G,f} = \frac{1}{2}LR_{G,i}$.

- Discriminador: $LR_{D,f} = \frac{1}{8}LR_{G,i}$.

Estableceuse `epoca_inicial = 20`, `epoca_final = 150` e `paso_epoca = 5` para determinar as épocas nas que se reduce a LR do xerador, e `epoca_inicial = 20`, `epoca_final = 120` e `paso_epoca = 5` para determinar as épocas nas que se reduce a LR do discriminador. Na Figura 5.1 amósase unha gráfica coa redución da LR do xerador e do discriminador durante un adestramento con 250 épocas.

No Cadro 5.6 numéranse do 29 ao 37 os adestramentos que se corresponden coas combinacións de hiperparámetros empregadas neste subexperimento. No Cadro 5.4 detállanse a relación entre a LR inicial e final para o xerador e o discriminador en cada un dos casos.

Adest.	$LR_{G,f}$	$LR_{D,f}$
ADA-1	$LR_{G,i}$	$0,25 \cdot LR_{D,i}$
ADA-2	$0,75 \cdot LR_{G,i}$	$0,125 \cdot LR_{D,i}$
ADA-3	$0,5 \cdot LR_{G,i}$	$0,125 \cdot LR_{D,i}$

Cadro 5.4: Relación entre a LR inicial e final para o xerador e o discriminador en cada un dos casos de redución da LR empregados no subexperimento relativo á LR-ADA.

5.1.4. Experimento 3: Avaliación con todos os conjuntos de datos

Finalmente, realizáronse 8 adestramentos adicionais para avaliar o modelo de clasificación con todos os conjuntos de datos multiespectrais presentados no capítulo anterior. Para isto, empregouse a mellor combinación de hiperparámetros obtida no experimento 2.

No anterior experimento, calculouse a OA e a AA a nivel de parche. Porén, para o experimento 3, calculouse a OA e a AA a nivel de píxel, co obxectivo de comparar os resultados obtidos coa ResBaGAN.

5.2. Descripción dos experimentos

Os experimentos leváronse a cabo na orde na que se describiron. En cada adestramento, gárdase un modelo en certos momentos do mesmo e avalíase utilizando as métricas descritas na Sección 4.2 co conxunto de validación. En cada adestramento, seleccionouse o modelo que obtivo unha maior AA co conxunto de validación. O motivo de empregar a AA en lugar da OA para seleccionar o modelo consiste en que a AA é unha métrica máis sensible ás clases minoritarias.

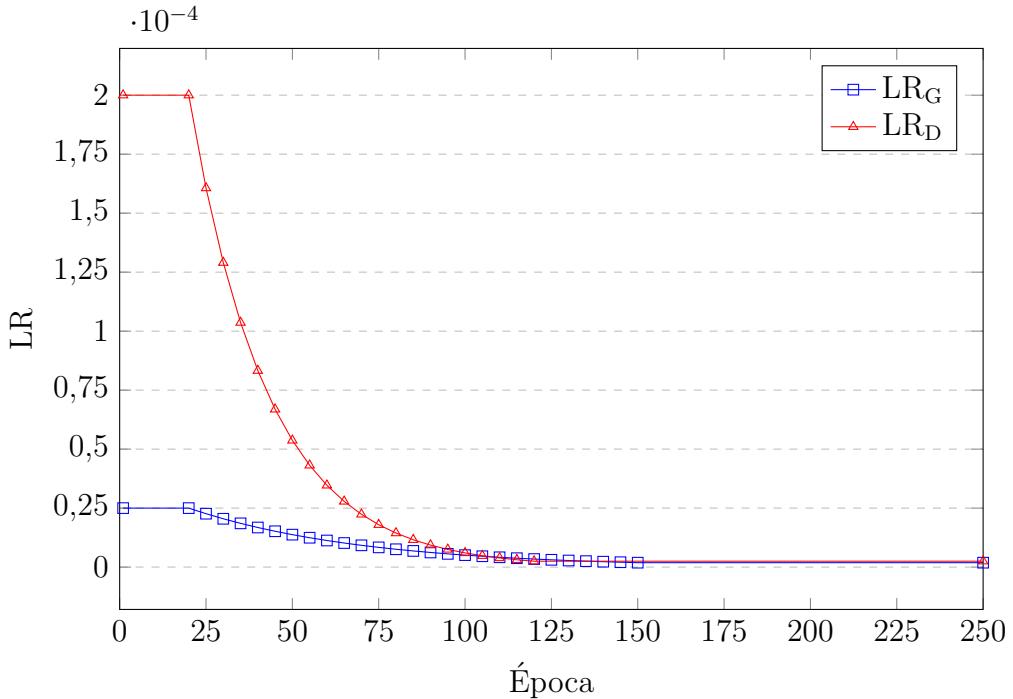


Figura 5.1: Redución da LR do xerador e do discriminador durante un adestramento con 250 épocas. Inicialmente, a LR do xerador é de $2,5 \cdot 10^{-5}$ e a do discriminador, de $2 \cdot 10^{-4}$. Entre a época 20 e a 150, a LR do xerador redúcese a $\frac{3}{4} \cdot 2,5 \cdot 10^{-5}$. Entre a época 20 e 120 a LR do discriminador redúcese a $\frac{1}{8} \cdot 2 \cdot 10^{-4}$.

Ademais, para cada modelo almacenado durante o adestramento, tamén se almacenaron 8 imaxes xeradas por clase. Deste modo, tamén se pode comparar a calidade das mostras xeradas polo modelo en cada adestramento. Na Sección 6 faranse algúns comentarios sobre a calidade destas.

5.3. Resultados

A continuación preséntanse os resultados obtidos en cada experimento. Os resultados do experimento 1 co conxunto de datos MNIST amósanse no Cadro 5.5. Na última columna indícase o índice do modelo seleccionado entre todos os que se gardaron en cada adestramento. Neste experimento, almacenouse un modelo cada 250 *batches*. Nas columnas 4 e 5 amósanse os valores de OA e AA obtidas co conxunto de validación co modelo seleccionado en cada adestramento.

No Cadro 5.5 obsérvanse que o mellor resultado foi obtido no adestramento 3, cunha OA do 95.52 % e unha AA do 95.42 % sobre o conxunto de validación. Sen realizar unha optimización de hiperparámetros extensa, obsérvase que a OA é inferior, pero aproxímase aos resultados conseguidos con outros modelos de

Experimento	Adest.	(LR _G , LR _D)	OA	AA	Modelo
Experimento 1:	1	($2 \cdot 10^{-4}$, $2 \cdot 10^{-4}$)	95.42	95.29	11/45
	2	($2 \cdot 10^{-4}$, $1 \cdot 10^{-4}$)	90.42	90.44	7/45
	3	($1 \cdot 10^{-4}$, $2 \cdot 10^{-4}$)	95.52	95.42	19/45
	4	($1 \cdot 10^{-4}$, $1 \cdot 10^{-4}$)	95.21	95.05	13/45

Cadro 5.5: OA e AA obtidas dos 4 adestramentos correspondentes ao experimento 1 sobre o conxunto de validación do conxunto de datos MNIST. O valor das columnas OA e AA correspón dese coa OA e tidas co modelo adestrado na época indicadoa na última columna. En **letra grosa** resálstanse os mellores resultados obtidos para cada métrica.

clasificación sobre este conxunto de datos amosados en [11]¹. Polo tanto, isto manifesta que a cStyleGAN2 pode aplicarse en problemas de clasificación de imaxes.

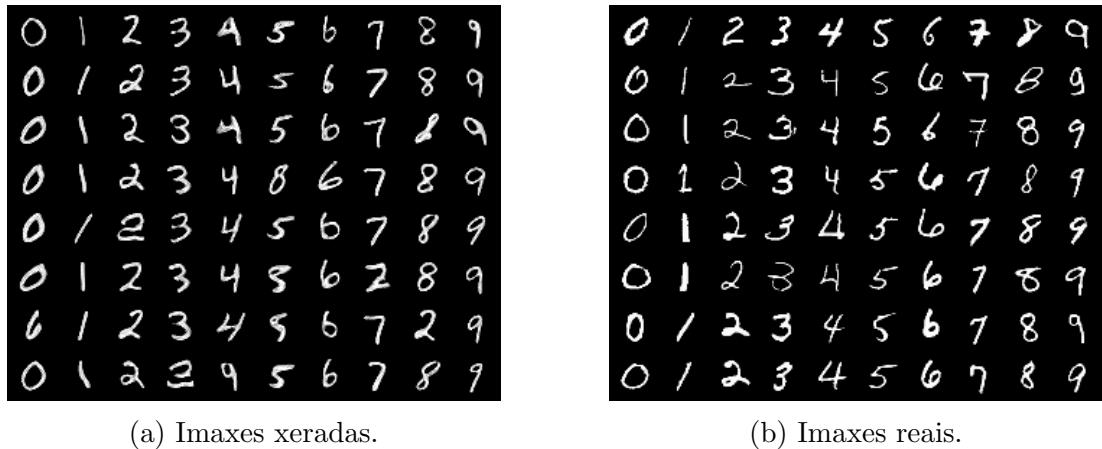


Figura 5.2: Imaxes xeradas ao final do adestramento 3 do experimento 1 co conxunto de datos MNIST. En ambas figuras, amósanse 8 imaxes por clase.

Na Figura 5.2a amósanse 80 imaxes xeradas ao final do adestramento 3 do experimento 1. Á dereita, na Figura 5.2b amosáse unha mostra das imaxes do conxunto de adestramento. Pódese concluír que as imaxes xeradas pola cStyleGAN co conxunto de datos MNIST son indistinguibles das reais. Ademais, apréciase que o modelo é capaz de producir imaxes condicionadas a unha etiqueta de clase específica. Aínda que hai algúns casos nos que o modelo produce unha imaxe dunha clase diferente á etiqueta de entrada (por exemplo, na séptima fila da Figura 5.2a, na primeira columna, xerouse un 6 cando debería xerarse un 0, e na penúltima

¹En [11] utilizan como métrica a *Taxa de erro* (ER), que se obtén como ER = 100 – OA.

columna, xerouse un 2 en lugar dun 8), na maioría dos casos o xerador consigue producir imaxes da clase correcta.

Os resultados do experimento 2 co conxunto de datos do río Oitavén amósanse no Cadro 5.6. Neste caso, a columna da dereita corresponde coa época na que se almacenou o modelo seleccionado para cada adestramento. Nas columnas 4 e 5 amósanse os valores de OA e AA a nivel de parche obtidas co conxunto de validación co modelo seleccionado en cada adestramento.

Unha das primeiras conclusións que se poden extraer a partir deste experimento é que se obteñen mellores resultados de clasificación se a LR do xerador é baixa e a do discriminador é alta. En particular, fixada a capacidade da rede a $C = 16$, cando a LR do xerador é maior ou igual que $1 \cdot 10^{-4}$ (adestramentos 5-12), a OA e a AA non superan o 80 %. As mellores configuracións para a LR do xerador e do discriminador con $C = 16$ resultaron ser as seguintes:

- En función da OA: $LR_G = 5 \cdot 10^{-5}$, $LR_D = 2 \cdot 10^{-4}$ (adestramento 13).
- En función da AA: $LR_G = 2,5 \cdot 10^{-5}$, $LR_D = 2 \cdot 10^{-4}$ (adestramento 17).

En relación á capacidade da rede, os resultados empeoran se se reduce C . Por outro lado, se se aumenta, a OA e a AA melloran nalgúns casos. Por exemplo, un caso no que melloran, aumentando a capacidade da rede de $C = 16$ a $C = 32$ e fixando $LR_G = 2,5 \cdot 10^{-5}$, $LR_D = 2 \cdot 10^{-4}$ (adestramentos 17 e 28, respectivamente), mellórarse a AA case nun 1 % e OA, máis dun 3 %.

En todos os adestramentos relativos a estes dous experimentos observouse un problema común: un sobreaxuste do discriminador. Na Figura 5.3 amósase a evolución da OA e da función de perda do discriminador en función das épocas para o adestramento 17. A partir da época 100, a función de perda do discriminador no conxunto de validación comeza a aumentar, mentres que a perda no conxunto de adestramento segue diminuíndo. A OA calculada sobre o conxunto de validación tamén comeza a diminuír a partir da época 150. O comportamento amosado nesta gráfica é común a todos os adestramentos realizados e, polo xeral, nos adestramentos cunha LR do xerador alta, o discriminador sobreaxusta antes.

Introducindo unha LR-ADA conseguiuse acadar mellores resultados no que respecta á OA e á AA. Ademais, comparando as Figuras 5.3 e 5.4 apréciase que, ao considerar unha LR-ADA, a OA calculada sobre o conxunto de validación non diminúa e a función de perda do discriminador aumenta menos. A mellor configuración de hiperparámetros para a LR-ADA, en función da AA, resultou ser a do adestramento 33.

Subexp.	Adest.	(LR_G, LR_D, C, ADA)	OA	AA	Modelo
LR _G e LR _D	5	(2 · 10 ⁻⁴ , 2 · 10 ⁻⁴ , 16, -)	64.78	76.93	48/250
	6	(2 · 10 ⁻⁴ , 1 · 10 ⁻⁴ , 16, -)	68.09	73.44	26/250
	7	(2 · 10 ⁻⁴ , 5 · 10 ⁻⁵ , 16, -)	47.36	59.96	34/250
	8	(2 · 10 ⁻⁴ , 2,5 · 10 ⁻⁵ , 16, -)	49.04	61.95	56/250
	9	(1 · 10 ⁻⁴ , 2 · 10 ⁻⁴ , 16, -)	76.5	79.97	54/250
	10	(1 · 10 ⁻⁴ , 1 · 10 ⁻⁴ , 16, -)	68.93	73.61	54/250
	11	(1 · 10 ⁻⁴ , 5 · 10 ⁻⁵ , 16, -)	43.69	68.43	46/250
	12	(1 · 10 ⁻⁴ , 2,5 · 10 ⁻⁵ , 16, -)	43.69	64.16	50/250
	13	(5 · 10 ⁻⁵ , 2 · 10 ⁻⁴ , 16, -)	83.95	84.88	86/250
	14	(5 · 10 ⁻⁵ , 1 · 10 ⁻⁴ , 16, -)	78.06	79.36	74/250
Capacidade da rede	15	(5 · 10 ⁻⁵ , 5 · 10 ⁻⁵ , 16, -)	68.69	76.35	66/250
	16	(5 · 10 ⁻⁵ , 2,5 · 10 ⁻⁵ , 16, -)	70.49	75.63	96/250
	17	(2,5 · 10 ⁻⁵ , 2 · 10 ⁻⁴ , 16, -)	79.87	85.56	106/250
	18	(2,5 · 10 ⁻⁵ , 1 · 10 ⁻⁴ , 16, -)	81.97	84.78	82/250
	19	(2,5 · 10 ⁻⁵ , 5 · 10 ⁻⁵ , 16, -)	78.55	83.59	114/250
	20	(2,5 · 10 ⁻⁵ , 2,5 · 10 ⁻⁵ , 16, -)	74.52	77.63	128/250
	21	(2 · 10 ⁻⁴ , 1 · 10 ⁻⁴ , 8, -)	58.23	66.54	34/250
	22	(2 · 10 ⁻⁴ , 1 · 10 ⁻⁴ , 32, -)	75.3	78.37	38/125
LR ADA	23	(1 · 10 ⁻⁴ , 1 · 10 ⁻⁴ , 8, -)	68.99	72.51	44/125
	24	(1 · 10 ⁻⁴ , 1 · 10 ⁻⁴ , 32, -)	71.27	82.03	46/125
	25	(5 · 10 ⁻⁵ , 1 · 10 ⁻⁴ , 8, -)	67.79	72.94	64/125
	26	(5 · 10 ⁻⁵ , 1 · 10 ⁻⁴ , 32, -)	82.39	84.63	78/250
	27	(2,5 · 10 ⁻⁵ , 1 · 10 ⁻⁴ , 8, -)	78.43	82.06	110/250
	28	(2,5 · 10 ⁻⁵ , 1 · 10 ⁻⁴ , 32, -)	82.99	86.44	98/250
	29	(5 · 10 ⁻⁵ , 2 · 10 ⁻⁴ , 16, ADA-1)	85.04	86.14	74/250
	30	(5 · 10 ⁻⁵ , 2 · 10 ⁻⁴ , 16, ADA-2)	82.99	86.17	88/250
	31	(5 · 10 ⁻⁵ , 2 · 10 ⁻⁴ , 16, ADA-3)	83.05	84.76	102/250
	32	(2,5 · 10 ⁻⁵ , 2 · 10 ⁻⁴ , 16, ADA-1)	85.22	86.60	110/250
	33	(2,5 · 10 ⁻⁵ , 2 · 10 ⁻⁴ , 16, ADA-2)	85.46	87.44	108/250
	34	(2,5 · 10 ⁻⁵ , 2 · 10 ⁻⁴ , 16, ADA-3)	85.10	86.67	182/250
	35	(2,5 · 10 ⁻⁵ , 1 · 10 ⁻⁴ , 32, ADA-1)	86.60	87.09	138/250
	36	(2,5 · 10 ⁻⁵ , 1 · 10 ⁻⁴ , 32, ADA-2)	86.00	87.04	142/250
	37	(2,5 · 10 ⁻⁵ , 1 · 10 ⁻⁴ , 32, ADA-3)	82.99	84.71	168/250

Cadro 5.6: OA e AA obtidas dos 30 adestramentos correspondentes ao experimento 2 sobre o conxunto de validación do conxunto de datos do río Oitavén. O valor das columnas OA e AA corresponde a OA e AA obtidas co modelo adestrado na época indicada na última columna. En **letra grosa** resálitanse os mellores resultados obtidos para cada subexperimento.

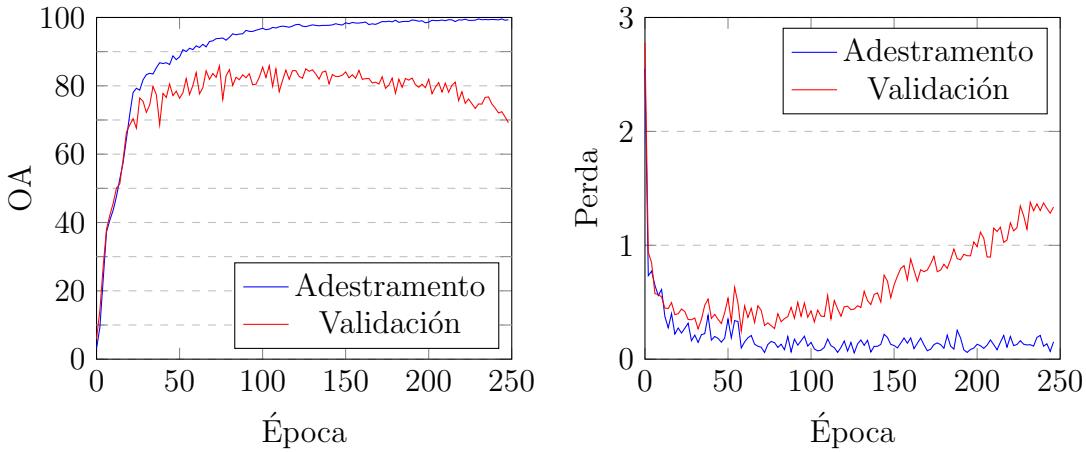


Figura 5.3: Evolución da OA e da función de perda do discriminador co conxunto de adestramento e de validación en función das épocas no adestramento 17 (ver Cadro 5.6).

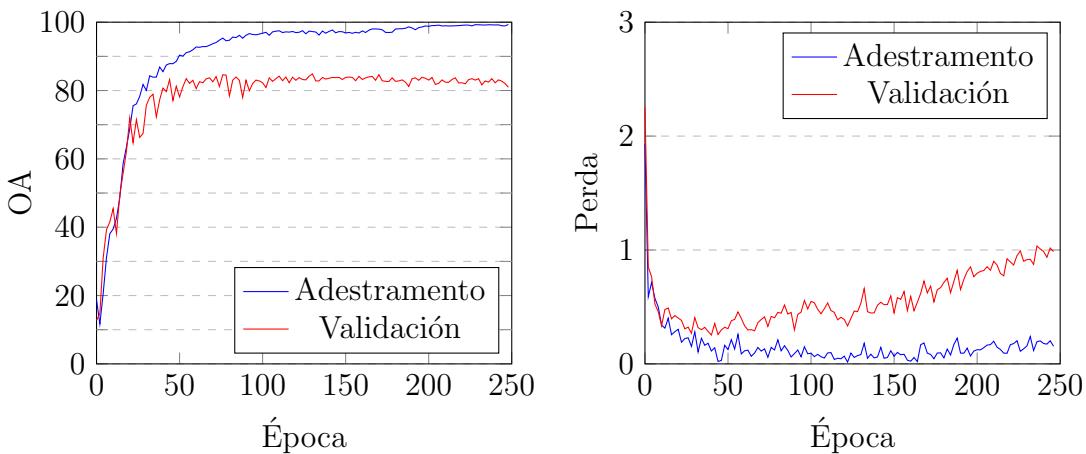


Figura 5.4: Evolución da OA e da función de perda do discriminador co conxunto de adestramento e de validación ao introducir unha LR-ADA en función das épocas no adestramento 33 (ver Cadro 5.6).

Na Figura 5.5a amósanse 80 imaxes xeradas ao final do adestramento 33 do experimento 2 co conxunto de datos do río Oitavén. Á dereita, na Figura 5.5b amósase unha mostra das imaxes do conxunto de adestramento. Realmente, tanto as imaxes reais como xeradas teñen 5 bandas, pero para poder visualizalas, amósanse as bandas RGB. Pode apreciarse que áinda que as imaxes xeradas presentan bordes mais difuminados, as texturas e cores correspondentes a cada clase se manteñen.

Os resultados do experimento 3 amósanse no Cadro 5.7. Os valores de OA e AA amosados correspóndense cos valores das métricas a nivel de píxel. As dúas

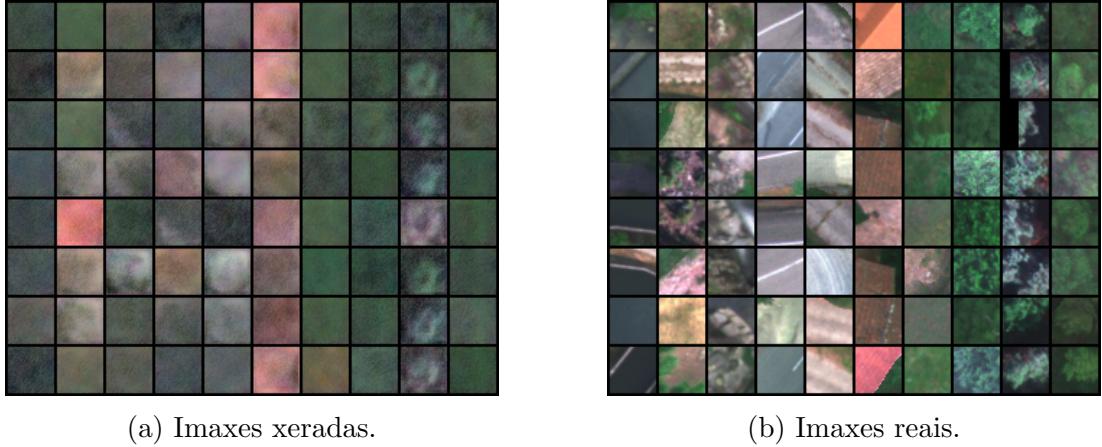


Figura 5.5: Imaxes xeradas ao final do adestramento 33 do experimento 2 co conxunto de datos do río Oitavén. En ambas figuras, amósanse 8 imaxes por clase.

primeiras columnas amosan os resultados presentados en [6] cunha CNN e coa ResBaGAN. A última columna amosa os resultados obtidos co modelo de clasificación proposto neste traballo.

No Cadro 5.7 obsérvase que a ResBaGAN obtén resultados moito mellores que a cStyleGAN2, nalgúns casos chegando a superar por 30 puntos porcentuais a OA ou a AA da cStyleGAN2. Isto sucede nos conxuntos de datos máis desbalanceados, como son o da cunca do Xesta, o do encoro de Eiras e o do río Mera. Unha das novidades que presenta a ResBaGAN (herdada da BaGAN [15]) precisamente é un mecanismo para lidar con conxuntos de datos desbalanceados. Este consiste en modelar as clases do problema de clasificación no espazo de características das imaxes tendo en conta a súa distribución no conxunto de datos orixinal. No conxunto do río Oitavén, que é o máis balanceado, a cStyleGAN2 consegue uns valores de OA e de AA más próximos aos da ResBaGAN.

En comparación coa CNN, de media a cStyleGAN2 obtén un resultado similar en canto á AA e un resultado claramente inferior en canto á OA. Posto que o discriminador da cStyleGAN2 consiste nunha CNN típica, estes resultados evidencian que o xerador da cStyleGAN2 non está contribuíndo a mellorar a clasificación, senón que está a introducir ruído.

Conxunto de datos		CNN	ResBaGAN	cStyleGAN2
Río Oitavén	OA	89.87	96.81	85.09
	AA	77.62	94.80	76.25
Cunca do Xesta	OA	96.35	98.24	87.31
	AA	67.30	88.23	55.98
Encoro de Eiras	OA	91.01	97.69	87.11
	AA	63.95	90.90	65.66
Regato Ermidas	OA	93.16	98.66	80.05
	AA	79.18	95.96	79.80
Río Ferreiras	OA	90.25	92.46	77.37
	AA	71.39	76.99	71.94
Río das Mestas	OA	89.36	91.50	84.15
	AA	79.15	87.03	80.29
Río Mera	OA	89.11	92.05	62.17
	AA	60.17	76.89	67.28
Río Ulla	OA	98.23	97.31	87.22
	AA	91.87	95.98	88.76
Media	OA	92.17	95.59	81.30
	AA	73.83	88.30	73.25

Cadro 5.7: Comparación dos resultados obtidos coa CNN, a ResBaGAN e a cStyleGAN2 para os conxuntos de datos multiespectrais presentados no capítulo anterior.

Capítulo 6

Discusión dos resultados

Neste capítulo discútense os resultados presentados na Sección 5.3. Isto inclúe a validación do modelo co conxunto de datos MNIST, os resultados acadados na optimización de hiperparámetros e a comparativa cos resultados acadados coa ResBaGAN e cunha CNN típica.

6.1. Validación do modelo

Os resultados do experimento co conxunto de datos MNIST amosan que a cStyleGAN2 pode aplicarse eficazmente en problemas de clasificación de imaxes, acadando unha alta exactitude sen unha extensa optimización de hiperparámetros. Ademais, as imaxes xeradas polo modelo son indistinguibles das reais e xeralmente corresponden á clase correcta, aínda que ocasionalmente prodúcense erros de clase.

A pesar de que probablemente se conseguisen mellores resultados de clasificación cunha optimización de hiperparámetros más extensa, o propósito deste primeiro experimento era validar a capacidade da cStyleGAN para clasificar imaxes e xerar imaxes condicionadas a unha etiqueta de clase específica. Polo tanto, os resultados acadados neste experimento son satisfactorios.

6.2. Optimización de hiperparámetros

Os resultados da optimización de hiperparámetros amosan que se obteñen mellores resultados de clasificación utilizando unha taxa de aprendizaxe baixa para o xerador e alta para o discriminador. Por outra banda, diminuir a capacidade da rede empora os resultados e, aumentala, mellóraos nalgúns casos.

Nos adestramentos nos que non se utilizou unha LR-ADA observouse que o discriminador tendía a sobreaxustar, especialmente cando a taxa de aprendizaxe do xerador era alta. A introdución dunha LR-ADA mellorou os resultados en termos de OA e AA e reduciu o efecto do sobreaxuste.

No que respecta ás imaxes xeradas, observouse que conseguén preservar a estrutura global da imaxe orixinal, pero non os detalles, como os bordes e as formas.

6.3. Comparación de resultados

Ao comparar os resultados acadados pola cStyleGAN2 cos acadados pola ResBaGAN e unha CNN típica presentados por Dieste et al. [6], observouse que a cStyleGAN2 non os consegue mellorar. Polo xeral, a cStyleGAN2 acada peores resultados nos conjuntos máis desbalanceados, sobre todo ao comparala coa ResBaGAN. Un dos posibles motivos polo existe tal diferenza consiste en que a ResBaGAN inclúe unha mellora con respecto a arquitectura típica das GAN á hora de representar as imaxes no espazo de características que permite facer fronte ao conjuntos de datos desbalanceados. Con todo, os valores de OA e AA acadados pola cStyleGAN2 co conjunto de datos do río Oitavén, o que presenta un menor desbalanceamento, están máis próximos aos conseguidos pola ResBaGAN.

Con respecto á comparación de resultados entre a cStyleGAN2 e a CNN, a cStyleGAN2 consegue obter un valor similar de AA de media e, nalgúns casos, superiores. Non obstante, atendendo á OA, a CNN acadou resultados significativamente superiores. Isto indica que o xerador da cStyleGAN2 non está contribuíndo a mellorar a clasificación, senón que está introducindo ruído.

Capítulo 7

Conclusións e posibles ampliacións

Neste traballo abordouse o problema da clasificación de imaxes multiespectrais de teledetección mediante unha arquitectura de cGAN baseada na StyleGAN2 disponible en [5]. Sobre a arquitectura de partida, modificouse o discriminador para que este poidese ser utilizado como clasificador.

Posteriormente, realizouse un estudo para determinar a configuración óptima dos hiperparámetros da rede para o problema de clasificación utilizando conxuntos de datos multiespectrais de teledetección de cuncas de ríos galegos.

Con todo isto, as principais aportacións deste traballo son as seguintes:

- Adaptouse a arquitectura StyleGAN2 condicionada ao problema de clasificación de imaxes multiespectrais de teledetección de alta resolución espacial correspondentes a ríos, e modificouse o discriminador para que puidese ser utilizado como clasificador.
- Comprobouse que StyleGAN2 condicionada presenta moi bo comportamento en canto á calidade de mostras xeradas sobre conxuntos de datos balanceados e con poucas variabilidade como son as imaxes RGB de números do conxunto MINIST. Comprobouse que tamén se produce un bo comportamento en canto a calidade visual das mostras xeradas no caso das imaxes multiespectrais de alta resolución de ríos, en que a variabilidade entre clases das imaxes é moi alta.
- Comprobouse que, aínda que a StyleGAN2 condicionada xera mostras de calidade visual aceptable e precisión de clasificación moi alta no caso das

imaxes multiespectrais, queda por debaixo de métodos especialmente diseñados para xerar mostras para problemas de clasificación con clases desbalanceadas como é a ResBaGAN.

Isto abre varias liñas de traballo futuro:

- Facer un estudo máis profundo sobre a influencia do desbalanceamento dos conjuntos de datos no rendemento da StyleGAN2 condicionada en problemas de clasificación e xeración de imaxes condicionadas a unha etiqueta de clase.
- Explorar diferentes modificacíons sobre a arquitectura da StyleGAN2 condicionada para mellorar o seu rendemento en problemas de clasificación, como as que presenta a ResBaGAN.

Apéndice A

Manuais técnicos

Neste apéndice comentaranse todas as instrucións necesarias para preparar un entorno en GNU/Linux sobre o que poder executar todo o código fonte involucrado na fase experimental deste traballo, de cara a facilitar a reproducibilidade dos resultados experimentais.

En primeiro lugar, cabe destacar que para poder executar o código é necesario dispoñer dunha GPU. Na Sección 4.3 proporcionase a especificación do entorno de probas no que se executaron os experimentos.

A.1. Adquisición do código

O código fonte da cStyleGAN2, así como os arquivos que facilitan a reproducibilidade dos experimentos pode ser consultado no seguinte repositorio de GitHub: <https://github.com/antongomez/Conditional-StyleGAN2/>. Ademais, tamén está dispoñible nunha carpeta en OneDrive¹.

A.2. Instalación de dependencias

Para facilitar o máximo posible a reprodución dos experimentos, xunto co código fonte facilítase un arquivo `environment.yml` que pode ser utilizado para crear un entorno de *Conda* coas dependencias necesarias para executar o código. Para iso, pódese executar o seguinte comando:

```
conda env create -f environment.yml
```

¹Ligazón: https://nubeusc-my.sharepoint.com/:f/g/personal/anton_gomez_lopez_rai_usc_es/Er0IZdnPf51Fhtt4PDuwZ2gBdv5yafd3xSWS5ao0qjFmnQ?e=3t3KHN

Para executar o comando é necesario ter instalado conda previamente. Pódese facer dende a ligazón: <https://conda.io/projects/conda/en/latest/user-guide/install/index.html>.

Unha vez creado o entorno conda, pódese activar con:

```
conda activate cstylegan2_conda
```

Unha vez creado e activado o entorno, dispone de todas as dependencias necesarias para executar o código.

A.3. Adquisición dos conxuntos de datos

Debido a que as imaxes multiespectrais de cuncas de ríos galegos empregadas neste traballo non son accesibles públicamente, inclúese na carpeta de OneDrive unha imaxe multiespectral resultante da redimensión e modificación de bandas dunha imaxe popular dentro da comunidade de teledetección: *Pavia University*. O Grupo de Intelixencia Computacional da Universidade do País Vasco pon a disposición do esta e máis imaxes: http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

Tanto na carpeta de OneDrive como no repositorio de GitHub, inclúese o conxunto de datos, de modo que non é necesario descargalo. De todos xeitos, cando se inicia un adestramento con este conxunto de datos e non está presente, o código descárgao automáticamente.

Apéndice B

Manuais de usuario

Neste apéndice indícanse as intruccións para repetir os experimentos realizados neste traballo. Previamente, é necesario ter creado e activado o entorno de conda tal e como se describiu no Apéndice A.

B.1. reptición dos experimentos

No directorio `cstylegan2` inclúense os ficheiros que implementan a rede. Para facilitar a interacción con eles, utilizáronse scripts de Bash onde se especifican os hiperparámetros de entrada da rede. Deste modo, dentro do directorio `scripts`, inclúense os seguintes ficheiros:

- `1_mnist.sh`: Script para reproducir os adestramentos co conxunto de datos MNIST (numerados do 1-4 no Cadro 5.5). Para executar o script, pódese emplegar o seguinte comando:

```
bash 1_mnist.sh MNIST
```

- `2_1_learning_rate.sh`: Script para reproducir os adestramentos co conxunto de datos OITAVEN variando a taxa de aprendizaxe do xerador e do discriminador (numerados do 5-20 no Cadro 5.6). Para executar o script, pódese emplegar o seguinte comando:

```
bash 2_1_learning_rate.sh OITAVEN
```

- `2_2_network_capacity.sh`: Script para reproducir os adestramentos co conxunto de datos OITAVEN variando a capacidade da rede (numerados do

21-28 no Cadro 5.6). Para executar o script, pódese emplegar o seguinte comando:

```
bash 2_2_network_capacity.sh OITAVEN
```

- **2_3_ada_learning_rate.sh**: Script para reproducir os adestramentos co conxunto de datos OITAVEN variando a taxa de aprendizaxe adaptativa (numerados do 28-37 no Cadro 5.6). Para executar o script, pódese emplegar o seguinte comando:

```
bash 2_3_ada_learning_rate.sh OITAVEN
```

- **3_all_datasets.sh**: Script para reproducir os adestramentos co con todos os conxuntos de datos (amosados no Cadro 5.7). Para executar o script, pódese emplegar o seguinte comando:

```
bash 3_all_datasets.sh
```

Todos os scripts, excepto o do experimento 3, requiren un argumento que é o nome do conxunto de datos co que se desexa adestrar a rede. Deste modo, podería levarse a cabo calquera dos experimentos con calquera dos conxuntos de datos. Cabe destacar que para reproducir os experimentos é necesario dispoñer dos conxuntos de datos e que se almacenen no directorio `data/NOME_CONXUNTO_DE_DATOS`.

Cada adestramento almacena os modelos no directorio `models/NOME_MODELO`, onde `NOME_MODELO` é un argumento que recibe a rede e que especifica o nome do adestramento. Dun xeito semellante, en cada adestramento almacénanse as imaxes xeradas no directorio `results/NOME_MODELO` e gárdanse os arquivos de *log* coas métricas de avaliación no directorio `logs/NOME_MODELO/`. Todos estes directorios pódense modificar no arquivo de configuración da rede: `cstylegan2/config.py`.

Por último, creouse un script de Python para avaliar a OA e AA a nivel de píxel da rede cos conxuntos de datos multiespectrais. Este script pódese executar co seguinte comando:

```
python cstylegan2/test_D.py --name NOME_MODELO \\
--folder NOME_CONXUNTO_DATOS \\
--model ID_MODELO
```

O argumento `ID_MODELO` é un número que identifica o modelo entre todos os modelos almacenados nun adestramento. Para coñecer cal é o modelo que presentou

unha maior AA entre todos os modelos almacenados, pódese executar o seguinte script de Python:

```
python scripts/extractBestModel.py --name NOME_MODELO
```

Ademais, creouse outro script de Python que xera gráficas para analizar visualmente as métricas que se almacenan ao longo de cada adestramento nos arquivos de *log*. Este script pódese executar co seguinte comando:

```
python scripts/genDataGraphics.py --name NOME_MODELO
```

B.2. Problemas frecuentes

Ás veces, a descarga automática do conxunto de datos MNIST non funciona correctamente. Nestes casos, pode descaragarse manualmente dende a carpeta de OneDrive onde se pon a disposición o código.

Apéndice C

Licenza

Este traballo está licenciado baixo a Licenza Creative Commons Atribución 4.0 Internacional (CC BY 4.0). Isto permite que outros compartan, adapten e usen o traballo, incluso con fins comerciais, sempre que se proporcione a atribución adecuada. Para máis detalles, consultar o texto da licenza.

Bibliografía

- [1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.
- [2] antongomez. Stylegan2 condicionada para clasificación. <https://github.com/antongomez/Conditional-StyleGAN2/>. Último acceso: 2024-09-13.
- [3] F. Argüello, D. B. Heras, A. S. Garea, and P. Quesada-Barriuso. Watershed monitoring in galicia from uav multispectral imagery using advanced texture methods. *Remote Sensing*, 13(14), 2021.
- [4] CESGA. Cesga technical documentation - tesla t4. https://cesga-docs.gitlab.io/ft3-user-guide/gpu_nodes.html#tesla-t4. Último acceso: 2024-06-26.
- [5] CIA-Oceanix. Conditional stylegan2 pytorch implementation. https://github.com/CIA-Oceanix/Conditional_StyleGAN2_pytorch. Último acceso: 2024-06-30.
- [6] Álvaro G. Dieste, Francisco Argüello, and Dora B. Heras. Resbagan: A residual balancing gan with data augmentation for forest mapping. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16:6428–6447, 2023.
- [7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Y. Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 3, 06 2014.
- [8] Zayd Mahmoud Hamdi, Melanie Brandmeier, and Christoph Straub. Forest damage assessment using deep learning on high resolution remote sensing data. *Remote Sensing*, 11(17), 2019.

- [9] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [10] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [11] Daniel Keysers. Comparison and combination of state-of-the-art techniques for handwritten character recognition: Topping the mnist benchmark, 2007.
- [12] Bing Lu, Phuong D. Dao, Jiangui Liu, Yuhong He, and Jiali Shang. Recent advances of hyperspectral imaging technology and applications in agriculture. *Remote Sensing*, 12(16), 2020.
- [13] lucidrains. Stylegan2 pytorch implementation. <https://github.com/lucidrains/stylegan2-pytorch>. Último acceso: 2024-06-30.
- [14] Vaïa Machairas, Matthieu Faessel, David Cárdenas-Peña, Théodore Chabardes, Thomas Walter, and Etienne Decencière. Waterpixels. *IEEE Transactions on Image Processing*, 24(11):3707–3716, 2015.
- [15] Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and Cristiano Malossi. Bagan: Data augmentation with balancing gan, 2018.
- [16] Giorgio Morales, Guillermo Kemper, Grace Sevillano, Daniel Arteaga, Ivan Ortega, and Joel Telles. Automatic segmentation of mauritia flexuosa in unmanned aerial vehicle (uav) imagery using deep learning. *Forests*, 9(12), 2018.
- [17] Pytorch. Mnist dataset. <https://pytorch.org/vision/stable/generated/torchvision.datasets.MNIST.html>. Último acceso: 2024-06-27.
- [18] Pytorch. Multistep lr. https://pytorch.org/docs/1.2.0/optim.html?highlight=multistep#torch.optim.lr_scheduler.MultiStepLR. Último acceso: 2024-06-27.
- [19] Daniel Scharvogel, Melanie Brandmeier, and Manuel Weis. A deep learning approach for calamity assessment using sentinel-2 data. *Forests*, 11, 11 2020.
- [20] Zhenfeng Shao, Penghao Tang, Zhongyuan Wang, Nayyer Saleem, Sarath Yam, and Chatpong Sommai. Brrnet: A fully convolutional neural network

- for automatic building extraction from high-resolution remote sensing images. *Remote Sensing*, 12(6), 2020.
- [21] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, July 2019.
 - [22] Vladimir N. Vapnik and Corinna Cortes. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
 - [23] Murong Wang, Xiabi Liu, Yixuan Gao, Xiao Ma, and Nouman Q. Soomro. Superpixel segmentation: A benchmark. *Signal Processing: Image Communication*, 56:28–39, 2017.
 - [24] McKell Woodland, John Wood, Brian M. Anderson, Suprateek Kundu, Ethan Lin, Eugene Koay, Bruno Odisio, Chung Caroline, Hyunseon Christine Kang, Aradhana M. Venkatesan, Sireesha Yedururi, Brian De, Yuan-Mao Lin, Ankit B. Patel, and Kristy K. Brock. Evaluating the performance of stylegan2-ada on medical images. In Can Zhao, David Svoboda, Jelmer M. Wolterink, and Maria Escobar, editors, *Simulation and Synthesis in Medical Imaging*, pages 142–153, Cham, 2022. Springer International Publishing.

Índice de figuras

1.1.	Exemplo da clasificación dunha imaxe MSI.	2
2.1.	Esquema dunha arquitectura típica dunha CNN para un problema de clasificación con 3 clases. A rede comeza cunha serie de capas convolucionais que extraen características das imaxes. A continuación, unha capa densa clasifica as imaxes en función das características extraídas.	6
2.2.	Esquema dun autoencoder. O codificador reduce a dimensión da imaxe de entrada a un vector de características, mentres que o decodificador reconstrúe a imaxe a partir do vector de características.	7
3.1.	Segmentación dunha imaxe en superpíxeles. Obsérvase como os superpíxeles agrupan píxeles con características similares.	10
3.2.	Extracción de parches a partir dunha imaxe MSI. En primeiro lugar realiza a segmentación en superpíxeles. A continuación, calcúlanse os centros dos superpíxeles calculando o cuadrilátero mínimo que os contén. Por último extráese un parche de 32×32 píxeles centrado no centro do superpíxel.	10
3.3.	Arquitectura dun xerador dunha GAN convencional (esquerda) e dun xerador da StyleGAN (dereita) [9].	12
3.4.	Transformación dun tensor constante nunha imaxe de 32×32 píxeles a través dos bloques do xerador da StyleGAN2. O tensor constante pasa por unha serie de bloques do xerador que aumentan progresivamente a súa resolución e reducen o número de canles. Despois de cada bloque, xérase unha imaxe intermedia da mesma resolución que o tensor mediante unha convolución. Esta imaxe intermedia combínase coa imaxe xerada no bloque anterior (tras realizar unha operación que duplica a súa resolución) mediante unha suma.	15

3.5. As capas convolucionais do discriminador dunha GAN típica transforman unha imaxe nun vector vector de características. Este vector de características é a entrada dunha capa densa que clasifica a imaxe como real ou falsa.	16
3.6. Hiperplano de separación das clases no discriminador da StyleGAN. Os parámetros do hiperplano veñen determinados polos pesos da neurona de saída da capa densa. As imaxes represéntanse no espazo a partir do vector de características obtido tras aplicar as capas convolucionais. As imaxes reais represéntanse mediante a cor azul e as imaxes sintéticas mediante a cor vermella. Os erros de clasificación represéntanse mediante unha cruz.	17
3.7. O discriminador da StyleGAN2 condicionada engade unha neurona de saída por cada clase do problema de clasificación. Cada unha destas neuronas especialízase en clasificar en reais ou falsas soamente as imaxes dunha das clases do problema.	20
3.8. Interpretación xeométrica da clasificación do vector de características \mathbf{x} (representado mediante un punto de cor azul) dunha imaxe real polo discriminador. Na esquerda, asignaráselle a \mathbf{x} a clase 1, pois o hiperplano con vector normal \mathbf{p}^1 é o que máis lonxe está de \mathbf{x} e que deixa a \mathbf{x} no semiplano negativo. Na dereita, todos os hiperplanos deixan a \mathbf{x} no semiplano positivo, polo que a clase de \mathbf{x} será a clase do hiperplano máis próximo a \mathbf{x} , neste caso, a clase 2.	22
4.1. Imaxes multiespectrais dos 8 ríos galegos en RGB (esquerda) xunto coa súa información de referencia (dereita). A clase correspondente a cada cor na información de referencia está descrita no Cadro 4.2, mentres que a cor negra significa que non hai información de referencia sobre devanditos píxeles.	25
5.1. Redución da LR do xerador e do discriminador durante un adestramento con 250 épocas. Inicialmente, a LR do xerador é de $2,5 \cdot 10^{-5}$ e a do discriminador, de $2 \cdot 10^{-4}$. Entre a época 20 e a 150, a LR do xerador redúcese a $\frac{3}{4} \cdot 2,5 \cdot 10^{-5}$. Entre a época 20 e 120 a LR do discriminador redúcese a $\frac{1}{8} \cdot 2 \cdot 10^{-4}$	36
5.2. Imaxes xeradas ao final do adestramento 3 do experimento 1 co conxunto de datos MNIST. En ambas figuras, amósanse 8 imaxes por clase.	37
5.3. Evolución da OA e da función de perda do discriminador co conxunto de adestramento e de validación en función das épocas no adestramento 17 (ver Cadro 5.6).	40
5.4. Evolución da OA e da función de perda do discriminador co conxunto de adestramento e de validación ao introducir unha LR-ADA en función das épocas no adestramento 33 (ver Cadro 5.6).	40

ÍNDICE DE FIGURAS 61

- | | |
|--|----|
| 5.5. Imaxes xeradas ao final do adestramento 33 do experimento 2 co
conjunto de datos do río Oitavén. En ambas figuras, amósanse 8
imaxes por clase. | 41 |
|--|----|

Índice de cadros

4.1. Descricións das imaxes multiespectrais das cuncas dos 8 ríos galegos. O número de superpíxeles etiquetados foi obtido a través de segmentación con WP.	24
4.2. Número de parches por clase en cada conxunto de datos. Os parches obtivéronse a partir da segmentación en superpíxeles con WP.	26
5.1. Esquema dos experimentos a realizar e numeración dos adestramentos que forman parte dos experimentos.	30
5.2. Valores da LR do xerador e do discriminador nos adestramentos co conxunto de datos MNIST.	31
5.3. Número de mostras de adestramento e validación para cada unha das clases presentes na imaxe do Río Oitavén para os adestramentos do experimento 2 (5-37).	32
5.4. Relación entre a LR inicial e final para o xerador e o discriminador en cada un dos casos de redución da LR empregados no subexperimento relativo á LR-ADA.	35
5.5. OA e AA obtidas dos 4 adestramentos correspondentes ao experimento 1 sobre o conxunto de validación do conxunto de datos MNIST. O valor das columnas OA e AA correspón dese coa OA e AA obtidas co modelo adestrado na época indicada na última columna. En letra grosa resáltanse os mellores resultados obtidos para cada métrica.	37
5.6. OA e AA obtidas dos 30 adestramentos correspondentes ao experimento 2 sobre o conxunto de validación do conxunto de datos do río Oitavén. O valor das columnas OA e AA correspón dese coa OA e AA obtidas co modelo adestrado na época indicada na última columna. En letra grosa resáltanse os mellores resultados obtidos para cada subexperimento.	39
5.7. Comparación dos resultados obtidos coa CNN, a ResBaGAN e a cStyleGAN2 para os conxuntos de datos multiespectrais presentados no capítulo anterior.	42