UNIVERSIDAD
DE GRANADA

جامعة الحسن الأول
UNIVERSITÉ HASSAN 1ER

DOCTORAL THESIS

LEADING TO THE AWARD OF THE FOLLOWING DOUBLE DEGREE:

DOCTORAT NATIONAL PAR L'UNIVERSITÉ HASSAN 1ER DE SETTAT (UH1), MOROCCO
AND
DOCTOR POR LA UNIVERSIDAD DE GRANADA (UGR), SPAIN

WITHIN THE FOLLOWING FRAMEWORK:

FORMATION DOCTORALE EN MATHÉMATIQUES APPLIQUÉES ET INFORMATIQUES MAI, CENTRE D'ETUDES DOCTORALES SCIENCES ET TECHNIQUES (UH1)
AND
THE PHD PROGRAM B25.56.1 INFORMATION AND COMMUNICATION TECHNOLOGIES (UGR)

Specialty : Computer Science

by

Yassir Benhammou

Preprocessing Techniques for more Robust Deep Learning Models: Application to Biomedical and Satellite Images

Thesis Supervision:

Pr.	Siham TABIK	Professor in University of Granada	Thesis Supervisor
Pr.	Boujemâa ACHCHAB	Professor in Hassan 1st University of Settat	Thesis Supervisor

Thesis Affiliations:

(**LAMSAD**) LABORATOIRE D'ANALYSE ET MODÉLISATION DES SYSTÈMES POUR L'AIDE À LA DÉCISION, UH1, BERRECHID 218, MOROCCO
And
(**DASCI**) THE ANDALUSIAN RESEARCH INSTITUTE IN DATA SCIENCE AND COMPUTATIONAL INTELLIGENCE, UGR, GRANADA 18071, SPAIN

Editor: Universidad de Granada. Tesis Doctorales
Autor: Yassir Benhammou
ISBN: 978-84-1117-678-1
URI: https://hdl.handle.net/10481/80011

Declaration of Authorship

This doctoral thesis entitled ”Preprocessing Techniques for more Robust Deep Learning Models: Application to Biomedical and Satellite Images” presented by Yassir Benhammou, has been carried out within the doctoral program ”B25.56.1 information and communication technologies” in the University of Granada under the supervision of Pr.Siham Tabik, and the doctoral program ”Mathématiques appliquées et informatiques” in the Hassan 1st University of Settat under the supervision of Pr.Boujemâa Achchab.

The PhD student Yassir Benhammou and the thesis supervisors Pr.Siham Tabik and Pr.Boujemâa Achchab, guarantee, by signing this doctoral thesis, the following:

- The work has been carried out by the PhD student under the direction of the thesis supervisors during his candidature for a PhD degree at both Universities.
- Not any part of this work has been previously submitted for a degree or any other qualification at both Universities or any other institution.
- To our knowledge, in the realization of this work, the rights of other authors to be cited have been respected when their results or publications have been used.

11^{th} of February 2022

Dedication

To my Dear Mother Latifa Sakat and my Dear Father Noureddine Benhammou,
to express my great love and my deepest
gratitude for all their sacrifices and amazing efforts,
spent for my education and my well-being
There is no dedication quite
enough to express my great love, my high regard and what
you deserve for your prayers and your great sacrifices, which
you have never ceased to provide me since my birth

To my dear Sisters Soukaina and Kawtar,
to express my best feelings of fraternity, my high
regard and my great love as recognition of your valuable
presence, your encouragement and your huge support

To Prof. Abdeljalil Sakat,
who was the first one to encourage me to do a PhD

To my whole Family

To all my Friends

To all my Professors, since the very first primary school until now,
who taught me and helped me arriving here

I dedicate this modest work.

Acknowledgments

Above all, I would like to thank God, my supreme creator for giving me the health, the strength, the guidance, the patience and the opportunity to achieve my dream in the best conditions.

Firstly, I would like to address my sincere gratitude to my supervisors Prof. Siham Tabik and Prof. Boujemâa Achchab for their support, their encouragements and their valuable guidance. I am very grateful for their valuable time spent with me during my research and the expertise they shared with me to reach this goal. Moreover, I would like to express my gratitude for their availability, their professional and personal qualities.
My acknowledgments go also for Prof. Domingo Alcaraz-Segura who helped me a lot during my thesis, introduce me to the Remote Sensing field and was always there with his support and guidance to improve my skills and reach new objectives in my research.
I'm very grateful and thankful for Prof. Francisco Herrera who believed in me, accepted me in his research group, supported my research and provided me with his wise guidance and valuable advice.
I would also like to express my gratitude to all researchers and colleagues that collaborated with me as co-authors during my thesis, who were always professional, ambitious, supportive, and helped us to reach new amazing perspectives together.
I would also like to express my gratitude and appreciation to both research entities that i belong to in Hassan 1st University and University of Granada, for their wonderful working conditions, their numerous activities and all the training sessions they provided me during my doctoral studies.
I would like to thank all my dear friends who suuported me during my thesis, i wish you a life full of happiness and success.
I am grateful to every person who contributed to the achievement of this work.

THANK YOU ALL !

Research funding

My Acknowledgment goes also to entities who have supported this thesis. This doctoral thesis has been supported:

- In the University of Granada by the program Proyecto de Investigación ”Sistemas inteligentes para problemas complejos usando Deep Learning” Ref. A-TIC-458.UGR18, Proyectos I+D+i del Programa Operativo FEDER 2018. Este proyecto está financiado por la Junta de Andalucía y al Fondo Europeo de Desarrollo Regional (FEDER).
- In Hassan 1st University by the National Grant of the Moroocan Ministry of Higher Education, Scientific Research and Executive Training.

Contents

List of Figures

List of Tables

General notations

Acronyms

AI	:=	Artificial Intelligence.
IT	:=	Information Technology.
HPC	:=	High Performance Computing.
DL	:=	Deep Learning.
ML	:=	Machine Learning.
CV	:=	Computer Vision.
ANN	:=	Artificial Neural Network.
CNN	:=	Convolutional Neural Network.
GAN	:=	Generative Adversarial Network.
RNN	:=	Recurrent Neural Network.
AE	:=	Autoencoder.
SAE	:=	Sparse Autoencoder.
SSAE	:=	Stacked Sparse Autoencoder.
DBN	:=	Deep Belief Network.
BM	:=	Boltzmann Machine.
RBM	:=	Restricted Boltzmann Machine.
DSN	:=	Deeply Supervised Net.
FC	:=	Fully Connected.
ReLUs	:=	Rectified Linear Units.
BN	:=	Batch Normalization.
TL	:=	Transfer Learning.
DA	:=	Data Augmentation.
IR	:=	Imbalance Ratio.
RGB	:=	Red, Green, Blue channels.
ILSVRC	:=	Image Large Scale Visual Recognition Challenge.
GPU	:=	Graphics Processing Unit.
CAD	:=	Computer-aided Diagnosis.
WSI	:=	Whole Slide Image.
ROI	:=	Region of Interest.
ILA	:=	Image Level Accuracy.
PLA	:=	Patient Level Accuracy.
AUC	:=	Area Under Curve.
H&E	:=	Hematoxylin and Eosin.
A	:=	Adenosis.
F	:=	Fibroadenoma.

PT	:=	Phyllodes Tumor.
TA	:=	Tubular Adenoma.
DC	:=	Ductal carcinoma.
LC	:=	Lobular Carcinoma.
MC	:=	Mucinous Carcinoma.
PC	:=	Papillary Carcinoma.
MSB	:=	Magnification-specific binary classification reformulation for BreakHis.
MIB	:=	Magnification-independent binary classification reformulation for BreakHis.
MSM	:=	Magnification-specific multi-category classification reformulation for BreakHis.
MIM	:=	Magnification-independent multi-category classification reformulation for BreakHis.
LBP	:=	Local Binary Patterns.
CLBP	:=	Completed Local Binary Pattern.
LPQ	:=	Local Phase Quantization.
GLCM	:=	Gray Level Co-Occurrence Matrices.
PFTAS	:=	Parameter-Free Threshold Adjacency Statistics.
ORB	:=	Oriented FAST and Rotated BRIEF.
RF	:=	Rotation Forest.
NN	:=	Nearest-neighbour.
K-NN	:=	K-Nearest Neighbor.
QDA	:=	Quadratic Linear Analysis.
RS	:=	Remote Sensing.
LULC	:=	Land Use/Land Cover.
gHM	:=	Global Human Modification.
CSV	:=	Comma-separated values.
tif	:=	Tagged Image File.
VMD	:=	Variational Mode Decomposition.
SVM	:=	Support Vector Machine.
LSSVM	:=	Least Squares Support Vector Machine.
SSVM	:=	Sparse Support Vector Machine.
LSVM	:=	linear Support Vector Machine
ASSVM	:=	Adaptive Sparse Support Vector Machine.
NPMIL	:=	Non-parametric Multi-instance learning.
CT	:=	Contourlet Transform.
PCA	:=	Principal Component Analysis.
GPR	:=	Gaussian Random Projection.
CBFS	:=	Correlation-Based Feature Selection.
FV	:=	Fisher Vector.
CSE	:=	Component Selective Encoding.
MIL	:=	Multiple Instance Learning.
MIP	:=	Multiple Instance Pooling.
L-ISOMAP	:=	LandMark ISOMAP.
DWT	:=	Discrete Wavelet Transform.
CBHIR	:=	Content-based Histopathological Image Retrieval.
XGB	:=	XGBoost classifier.
MDT	:=	Meta-decision Tree.
FAO	:=	Food and Agriculture Organisation.
WV2	:=	WorldView-2 Dataset.
WV3	:=	WorldView-3 Dataset.
TP	:=	True Positive.

FN	:=	False Negative.
TN	:=	True Negative.
FP	:=	False Positive.
F1	:=	F1-score.
GEE	:=	Google Earth Engine.
WRS	:=	Wilcoxon Rank Sum.
MNN	:=	Multilayer Neural Network.
FD	:=	Fractal dimension.
Ent	:=	Entropy.
PWT	:=	Pyramid-Structure Wavelet Transform.

Pre- and post-processing methods abbreviations

Res(h x w)	:=	Resizing original images to a new size (h x w).
SMI	:=	Subtracting mean images.
SN	:=	Stain normalization.
SN(x)	:=	Stain normalization using method x.
UP	:=	Minority class upsamling.
RGBT	:=	RGB channels information transformation.
NDS	:=	Nuclei Detection and Segmentation using region growing technique.
ETB	:=	E_AHE and TB_HAT techniques.
GSC	:=	Grey-scale conversion.
MVD	:=	Multilevel variational mode decomposition.
KM	:=	K-mean clustering.
CE	:=	Contrast Enhancement.
JPEG	:=	Conversion to JPEG.
CD	:=	Color distortion.
HP	:=	Hue permutation.
BS	:=	Brightness saturation.
Bin	:=	Binarization.
DA(x)	:=	Data augmentation using x.
DAB(x)	:=	Data augmentation with a data balancing purpose using x.
IV	:=	Intensity variation.
Rot	:=	Rotation.
Tr	:=	Translation.
Flip	:=	Flipping.
Scal	:=	Scaling.
Mir	:=	Mirroring.
Dis	:=	Distortion.
Zoom	:=	Zooming.
CROP	:=	Cropping.
ANP	:=	Adding noisy points.
HS	:=	Height shift.
WS	:=	Width shift.
Res	:=	Resizing.
P(a x a)	:=	Patches extraction of size (a x a).
Rnd(a x a)	:=	Random patches extraction of size (a x a).

SW(a x a)	:=	Sliding window patches extraction of size (a x a).
MKV(N)	:=	Extraction of N patches using MKV framework.
SQ	:=	Division into non-overlapping square tiles.
JCTF	:=	Joint color-texture features extraction.
Zer	:=	Zernaike moments.
HI	:=	Histogram information extraction.
KAZE	:=	KAZE features extraction.
Tam	:=	Tamura features extraction.
BE	:=	Binarization encoding.
FV(x)	:=	Fisher vector encoded using a model x.
DR(f,x)	:=	Dimentionality reduction of features f using a method x.
Rlf	:=	Relief.
PROJ	:=	Projection into an invariant space.
LR	:=	Logistic Regression.
Integrated	:=	A model combining the best features extractors and classifiers at each magnification level.
SBC	:=	Similarity based comparison using Hamming distance.
NDCNN	:=	A new designed CNN model.
NDCNN(x)	:=	A new designed CNN model inspired from x.
NDCNN(x,trans)	:=	A new designed CNN model constituted of x with an integrated transition layer.
E(C)	:=	Ensemble of classifiers C.
Eiter(C)	:=	Ensemble of the same classifier C outputs captured at different iterations of its training.
Emv(C)	:=	Ensemble of classifiers C using majority voting rule.
Eavg(C)	:=	Ensemble of classifiers C using average rule.
Esum(C)	:=	Ensemble of classifiers C using sum rule.
Emax(C)	:=	Ensemble of classifiers C using max rule.
Emdt(C)	:=	Ensemble of classifiers C using MDT.
ImageNet	:=	The used model was pretrained on ImageNet.
Camelyon	:=	The model was pre-trained on Camelyon16 dataset images.
Im-Break	:=	The used CNN that was fine-tuned on the BreakHis multi-category classification task.

Land use and Land cover classes dictionary

UrbanBlUpArea	:=	Urban .
BarrenLands	:=	Barren.
MossAndLichen	:=	Moss and Lichen.
SrublandClose	:=	Close Shrublands.
ShrublandOpen	:=	Open Shrublands.
WetlandMarshl	:=	Marshland.
WetlandSwamps	:=	Swamp.
WetlandMangro	:=	Mangrove.
Grasslands	:=	Grassland.
CropBroadRain	:=	Rainfed Broadleaf Cropland.
CropBroadIrri	:=	Irrigated Broadleaf Cropland.
CropCereaRain	:=	Cereal Rainfed Cropland.
CropCereaIrri	:=	Cereal Irrigated Cropland.
CropSeasWater	:=	Cropland Seasonal Water.
ForestsDeEvNe	:=	Dense Evergreen Needleleaf Forest.

ForestsClEvNe	:=	Close Evergreen Needleleaf Forest.
ForestsOpEvNe	:=	Open Evergreen Needleleaf Forest.
ForestsDeEvBr	:=	Dense Evergreen Broadleaf Forest.
ForestsClEvBr	:=	Close Evergreen Broadleaf Forest.
ForestsOpEvBr	:=	Open Evergreen Broadleaf Forest.
ForestsDeDeNe	:=	Dense Deciduous Needleleaf Forest.
ForestsClDeNe	:=	Close Deciduous Needleleaf Forest.
ForestsOpDeNe	:=	Open Deciduous Needleleaf Forest.
ForestsDeDeBr	:=	Dense Deciduous Broadleaf Forest.
ForestsClDeBr	:=	Close Deciduous Broadleaf Forest.
ForestsOpDeBr	:=	Open Deciduous Broadleaf Forest.
PermanentSnow	:=	Permanent Snow.
WaterBodyCont	:=	Continental Water Bodies.
WaterBodyMari	:=	Marine Water Bodies.

Abstract

Computer Vision (CV) is an Artificial Intelligence (AI) field that replicate the human eyes and brain's ability in perceiving images and understanding them. Deep learning (DL) models and especially Convolutional Neural Networks (CNNs) have become the state-of-the-art in most complex CV tasks. These models learn automatically to take decisions based on imagery data without being explicitly programmed for this purpose as it is the case in self-driving cars or smartphones face recognition systems.

CNNs consist in a huge number of interconnected Artificial Neural Networks (ANNs) with trainable parameters widely inspired from the way the human brain neurons learn and transmit knowledge to each other. Hence, training them for a specific task requires a large number of carefully annotated images. However, for complex problems, such as those addressed in this thesis, creating high quality training datasets is very expensive, requires a high level of expertise and a huge amount of work. To overcome these limitations, the main adopted techniques in the literature are data preprocessing and Transfer Learning (TL). In the latter, CNNs are firstly pretrained on available large natural images datasets such as ImageNet, then retrained on target domain datasets containing less images. Whereas, data preprocessing involves all the transformations applied to datasets in order to improve their size and value. In this thesis, we proposed preprocessing techniques to improve the robustness of DL models in two complex applications: biomedical and satellite images classification.

In the first application, we combined the state-of-the-art CNN, the most adequate data preproccesing and transfer learning methods with the benchmark dataset used in that problem called BreakHis, to elaborate an ideal automatic system for breast cancer diagnosis from both clinical and technical standpoints. And our analysis has demonstrated that the complexity of this problem related to its data quality and annotation, hugely affect the performance of the trained DL model even in a well built methodological approach. In the second use case, we trained DL models on our own built dataset for automatic Land Use/Land Cover (LULC) classification. To our knowledge, the dataset we proposed called Sentinel2LULC, is the largest global high resolution and free satellite images dataset adapted for DL usage in this problem. This dataset was carefully built using the big amount of remote sensing data available nowadays on free platforms such as Google Earth Engine and a carefully designed methodology to transform all these data into a high value dataset for this specific problem. The experimental analysis in conjunction with DL models in this second scenario has achieved very promising results and proved the dataset quality importance. The particular conclusion in each one of these two studies allowed us to build our main conclusion of this thesis: even when the state-of-the-art models and methods are adopted and combined, the data quality remains the major source gold for CNNs training and constitute the key factor to reach a good performance in complex CV tasks.

Keywords : Deep Learning, Machine Learning, Computer Vision, Convolutional Neural Networks, Image Classification, Data Preprocessing, Transfer Learning, Breast cancer, Biomedical Imagery, Remote sensing, Land use, Land cover, Satellite Imagery.

Résumé

La vision par ordinateur (CV) est un domaine de l'intelligence artificielle (AI) qui reproduit la capacité des yeux et du cerveau humains à percevoir les images et à les comprendre. Les modèles d'apprentissage profond (DL), et en particulier les réseaux de neurones convolutifs (CNNs), sont devenus l'état de l'art pour les tâches du CV les plus complexes. Ces modèles apprennent automatiquement à prendre des décisions en utilisant un ensemble d'images sans être explicitement programmés pour cette fin, notamment dans les voitures à conduite autonome ou les systèmes de reconnaissance faciale des smartphones.
Les CNNs sont constitués d'un grand nombre de réseaux neuronaux artificiels (ANNs) avec des paramètres entraînables inspirés de la façon dont les neurones du cerveau humain apprennent et se transmettent les connaissances entre eux; et leur entraînement nécessite un grand nombre d'images soigneusement annotées. Cependant, pour des problèmes complexes, tels que ceux abordés dans cette thèse, la création d'un ensembles d'images d'entraînement de haute qualité est très coûteuse et exige une haut expertise. Pour surmonter ces limitations, les principales techniques adoptées dans la littérature sont le prétraitement des données et l'apprentissage par transfert. Dans ce dernier, les CNNs sont d'abord pré-entraînés sur de grands ensembles d'images naturelles tels que ImageNet, puis ré-entraînés sur des ensembles du domaine cible contenant moins d'images. Alors que le prétraitement des données implique toutes les transformations appliquées pour augmenter la taille et améliorer la valeur des données. Dans cette thèse, on propose des techniques de prétraitement pour améliorer la robustesse des modèles du DL dans deux applications complexes : la classification d'images biomédicales et d'images satellitaires.
Dans la première application, nous avons combiné avec l'ensemble de données appelé BreakHis, l'état de l'art CNN, ainsi que les méthodes de prétraitement et d'apprentissage par transfert les plus adéquates, afin de construire un système automatique idéal pour le diagnostic automatique du cancer du sein, tant du point de vue clinique que technique. Notre analyse a démontré que la complexité de ce problème, liée à la qualité de ses données, affecte considérablement la performances du DL, même avec méthodologie bien construite. Dans le deuxième cas d'utilisation, nous avons entraîné des modèles DL sur notre propre ensemble de données appelé Sentinel2LULC pour la classification automatique de l'utilisation et couverture des sols. À notre connaissance, Sentinel2LULC est le plus grand ensemble d'images satellitaires à échelle mondiale, à haute résolution et gratuit adaptées au DL. il a été soigneusement construit à partir de la grande quantité de données de télédétection disponibles aujourd'hui sur les plateformes gratuites telles que Google Earth Engine avec une méthodologie soigneusement conçue pour résoudre ce problème. L'analyse des modèles DL dans ce deuxième scénario a aboutit à des résultats prometteurs et a prouvé l'importance de la qualité des données.
La conclusion particulière de chacune de ces deux applications nous a permis de formuler notre conclusion principale de cette thèse : même lorsque les modèles DL et méthodes de pointe sont adoptés et combinés, la qualité initiale des données reste le facteur le plus imporatnt pour atteindre une bonne performance dans les tâches complexes du CV.

Resumen

La visión por ordenador (CV) es un campo de la Inteligencia Artificial (AI) que replica la capacidad de los ojos y el cerebro humanos para percibir imágenes y comprenderlas. Los modelos de aprendizaje profundo (DL) y, en especial, las redes neuronales convolucionales (CNNs) se han convertido en el estado del arte en las tareas más complejas de CV. Estos modelos aprenden automáticamente a tomar decisiones en función de los datos sin necesidad de ser programados explícitamente para ello, como ocurre en los coches autoconducidos o en los sistemas de reconocimiento facial de los smartphones.

Las CNNs consisten en un gran cantidad de redes neuronales artificiales (ANNs) interconectadas con parámetros entrenables inspirados de la forma en que las neuronas del cerebro humano aprenden y se transmiten conocimientos. Por lo tanto, entrenarlas para una tarea específica requiere un gran cantidad de imágenes cuidadosamente anotadas. Sin embargo, para problemas complejos, como los que se abordan en esta tesis, la creación de datos de entrenamiento de alta calidad es muy cara y requiere un alto nivel de experiencia. Para superar estas limitaciones, las principales técnicas adoptadas en la literatura son el preprocesamiento de datos y el aprendizaje por transferencia (TL). En este último, las CNNs se preentrenan primero en grandes conjuntos de datos de imágenes naturales como ImageNet, y luego se reentrenan en datos del dominio de destino. Por su parte, el preprocesamiento de datos implica todas las transformaciones aplicadas a los datos para mejorar su tamaño y valor. En esta tesis, propusimos técnicas de preprocesamiento para mejorar la robustez de modelos DL en dos aplicaciones complejas: la clasificación de imágenes biomédicas y de satélite.

En la primera aplicación, combinamos la CNN de última generación, los métodos de preprocesamiento y de aprendizaje de transferencia más adecuados con el conjunto de datos de referencia utilizado en ese problema llamado BreakHis, para elaborar un sistema automático ideal para el diagnóstico del cáncer de mama tanto desde el punto de vista clínico como técnico. Y nuestro análisis ha demostrado que la complejidad de este problema relacionada con la calidad de sus datos y su anotación, afecta enormemente al rendimiento del modelo DL entrenado incluso en un enfoque metodológico bien construido. En el segundo caso, entrenamos modelos DL con nuestro propio conjunto de datos para la clasificación automática del uso y la cobertura del suelo (LULC). Hasta donde sabemos, el conjunto de datos que propusimos, llamado Sentinel2LULC, es el mayor conjunto de datos global de imágenes de satélite de alta resolución y gratuitas adaptado para el uso de DL. Este conjunto de datos fue cuidadosamente construido utilizando la gran cantidad de datos de teledetección disponibles hoy en plataformas gratuitas como Google Earth Engine (GEE) y una metodología cuidadosamente diseñada para transformar todos estos datos en un conjunto de datos de alto valor. El análisis experimental con los modelos DL en este segundo escenario ha logrado resultados muy prometedores y ha demostrado la importancia de la calidad de datos. La conclusión particular en cada uno de estos estudios nos permitió construir nuestra conclusión principal de esta tesis: incluso cuando se adoptan y combinan los modelos y métodos más avanzados, la calidad de los datos sigue siendo el factor clave para alcanzar un buen rendimiento en tareas complejas de CV.

CHAPTER 1

Introduction

Contents

1.1 Research context and preliminaries

1.1.1 General context

1.1.1.1 Emergence of Deep Learning and its data-related limitations

Today, AI has become a priority in most countries around the world, at both economical and educational levels, and constitutes one of the most important levers for human development. Thus, every week, top of the notch Information Technology (IT) companies and their AI research laboratories including: Google, Microsoft, Facebook, Apple, Amazone, etc produce a huge amount of new research papers, models, datasets, frameworks and computing resources to ease AI usage for practitioners in the research community. This serious willingness to popularize AI concepts comes from the outstanding breakthroughs achieved recently by the core component of AI: Machine Learning (ML). ML is the most popular branch of AI that allows computers to be as smart as humans or even outperform them in solving several complex problems. In ML, computers learn automatically to make predictions and take decisions from available data in a given problem without being explicitly programmed for this specific purpose. In fact, ML is the main used technique to convert data into knowledge understandable by machines. Mathematics and statistics are the fundamental foundation of ML, and together consist the basis behind ML algorithms ability of analyzing large amounts of data, identifying their patterns and deduce the most important features inside these patterns.

Reaching this current era where intelligence is transmitted from humans to machines was not easy, and both humans and computers went together through many important steps in the history of AI to achieve this goal. The story started in 1950 when the mathematician Alan Turing was wondering either machines could also think as humans do or not. Then, several scientists followed this research line and tried to artificially replicate the human brain behaviour for knowledge acquisition. After many attempts and a long period of intensive research, the real breakthroughs started at the end of the 20th Century with the appearance of internet, the availability of Big Data and High Performance Computing (HPC) processors. Alongside with these computer-side breakthroughs, researchers brought new reformulations to intelligent machine conception and started to propose new ML models widely inspired from the human brain structure. In fact, deep ANNs replicate closely the way human neurons learn and transmit information to each other through the interconnected network of brain nodes. Nowadays, these models are referred to with ”DL” or ”deep ANNs”; although, original shallow ANNs were proposed before the end of the 20th century but the available amount of data, the computing power and training strategies weren't sufficient enough to make it reach the desired performance.

In general, DL models are trained and learn from input data throughout the continuous adaptation of their interconnected nodes. The latter are trainable weights and bias values organized into stacked layers connected with mathematical transformation functions. In fact, DL models are an improved version of shallow ANNs in terms of the amount of layers, nodes and other parameters that characterize the ANN depth. In fact, due to their very large number of trainable parameters, DL models are taking more advantage of large datasets and the computing power available nowadays to reach a considerable performance. In addition, as DL includes multiple architectures like CNNs, Deep Belief Networks (DBNs), Generative Adversarial Networks (GANs), Recurrent Neural Networks (RNNs), etc. it can be used in conjunction with different types of real-world data types (images, sound, text, time series, etc.) to perform prediction tasks in various AI applications including CV, audio/speech recognition, machine translation, recommendation systems, social network filtering, bioinformatics, RS and so much more.

In this Thesis, we focus on two of the most important and complex CV applications which are biomedical and satellite images classification. In concrete terms, CV is an AI technique that allows computers to perform a human-alike analysis of images captured by a camera or other imagery sensing devices in order to understand and recognize their content automatically. DL models and especially CNNs have shown to be very efficient or even exceed humans in recognizing an image, understanding it, and processing its content. CNNs are DL models that are inspired from the human eyes and brain's ability to process and analyze perceived images. Since the first day when these models have proven themselves, practitioners are continuously trying to delegate them the most complex vision-based human tasks. Generally, for image classification, CNNs requires a big amount of training samples that should be images annotated by expert in each domain of application. However, collecting high quality annotated images for this kind of applications is very expensive, requires an enormous work a high level of expertise in the domain of application. To overcome these limitations, data preprocessing and TL have emerged as the reference techniques.

1.1.1.2 Used techniques in Deep Learning to overcome data-related issues

- **Data preprocessing** Data preprocessing involves all the transformations applied to raw datasets in order to prepare them for further analysis. In fact, creating a dataset is a human task and as all human tasks, it is prone to errors and other factors that introduce noise and inconsistencies to these data. Thus, the majority of datasets suffers from missing values, redundancies, noise or even an imbalance ratio between their classes due to the difficulty of collecting sufficient samples in certain problems. Notably, in biomedical imagery datasets related to a certain disease, we often face a data imbalance issue. This problem is mainly due to the fact that during data collection, the nature of a disease and its occurrence rate impose on us to have less positive than negative samples. Therefore, data preprocessing phase includes data cleaning, data balancing, data augmentation, normalization, dimension reduction, noise reduction and many other operations that will help the DL model to reach a better outcome. Another limitation that faces these kind of datasets is the insufficient number of contained samples. And giving the fact that DL requires a very large number of data in order to achieve good classification results, practitioners apply what we call Data Augmentation (DA) with the aim to artificially increase the size of these datasets. Generally, to augment a dataset size, distortion, rotating, flipping, mirroring, zooming in/out as long as other more complicated transformations can be applied to raw images to create new one that belongs to the same classes as the original ones. Preserving the original annotation after DA is easily understandable, but what is more interesting is to know that CNNs are rotation-invariant, flipping-invariant and so on for all other transformations. This characteristic reveals another interesting analogy between CNNs and the human eye which is also able to recognize visual features in different positions and conditions.

- **Transfer Learning(TL)** In TL, CNNs are pretrained on recently available natural images datasets containing millions samples annotated with thousands classes (e.g., ImageNet), then further trained on the specific target dataset containing images with labels related to the application of interest, which often have less samples than the first pretraining dataset. The intuition behind this practice comes from the fact that the pretraining step helps the network learn general features shared between images from different domains, and reuse them on the specific target task. The solution brought by this two-stage approach to the major problem of AI related to data availability has attracted the big pioneers in AI, including Google who started to propose ready-to-use ImageNet pretrained CNN models. In addition to that, they offered open source frameworks such as TensorFlow and Keras that ease the implementation of these pretrained models under different environments and adapted them for all kind of computing platforms ranging from expensive HPC servers to relatively affordable and efficient Graphics Processing Units (GPUs).

1.1.2 Deep Learning models

as DL includes multiple architectures like CNNs, Deep Belief Networks (DBNs), Generative Adversarial Networks (GANs), Recurrent Neural Networks (RNNs), etc. it can be used in conjunction with different types of real-world data types (images, sound, text, time series, etc.) to perform prediction tasks in various AI applications including CV, audio/speech recognition, machine translation, recommendation systems, social network filtering, bioinformatics, RS and so much more.

1.1.2.1 Convolutional Neural Networks (CNNs)

Nowadays, CNNs have clearly outperformed all classical handcrafted based models in several CV benchmark challenges [143]. Notably, Image Large Scale Visual Recognition Challenge (ILSVRC) for ImageNet [38] dataset classification. In 2012, AlexNet, based on the well known LeNet, achieved outstanding results on this dataset composed of 1.2 million images distributed in 1000 different classes [93].

Since then, CNN based models have been leading the first place in this competition and continue to show an important potential with a large room of improvement.

A CNN model is the concatenation of N layers between the input and the output layers trained in an end-to-end fashion, in order to learn the hidden function $M()$ that maps each input image X to its correct class label Y among K outputs. This function calculates a score probability of each image in regards to each class through the composition of different layer functions L_i with $(1 \leq i \leq N)$ as depicted in the definition below and figure 1.1:

$$M(X) = L_N(L_{N-1}(...(L_1(X)))) \tag{1.1.1}$$

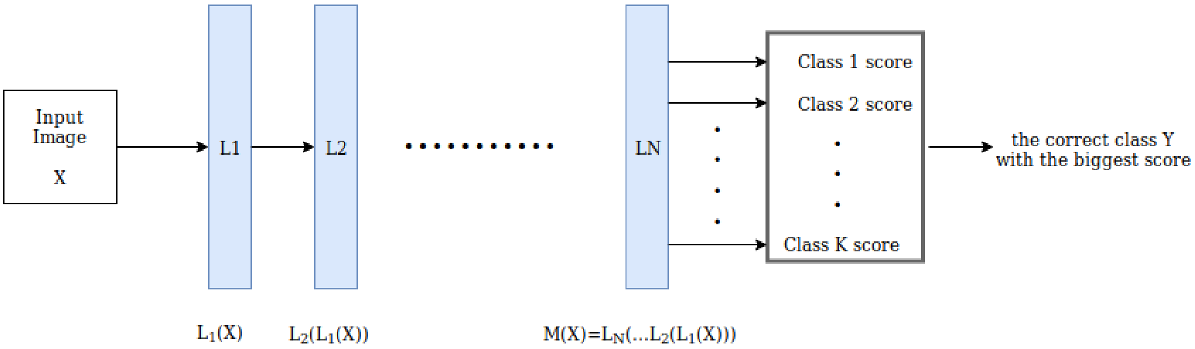


Figure 1.1: A typical CNN architecture composed of different layers.

Among all DL models, CNNs are the most adequate to deal with high dimension images. In fact, CNNs acts like a long dimensionality reduction process binding input images to their classification scores outputs, along with a learning process of the most significant features for each class. A CNN layer could be either a convolutional layer, a pooling layer or a fully connected layer:

- **Convolutional layer** this layer is the core component of CNNs. Often, these models are composed of several convolutional layers placed at different network depth levels. For each convolutional layer L, with an input tensor FM_{L-1} of three dimensions $(x1 \times y1 \times z)$ containing obtained feature map from predecessor layer, and a set of weights W_L called also filter bank [95] composed of k weight filters with a receptive field of size $(r \times r \times z)$ each, in addition to a bias b_L. We have an output feature map FM_L for each depth slice j computed as follow and illustrated in figure 1.2:

  $$FM_L(j) = f(\Sigma_{i=1}^{i=z} FM_{L-1}(i) * W_L(i,j) + b_L(j)) \tag{1.1.2}$$

  where $f()$ is the activation function at layer L and $(*)$ denotes a discrete convolution operation between two tensors.

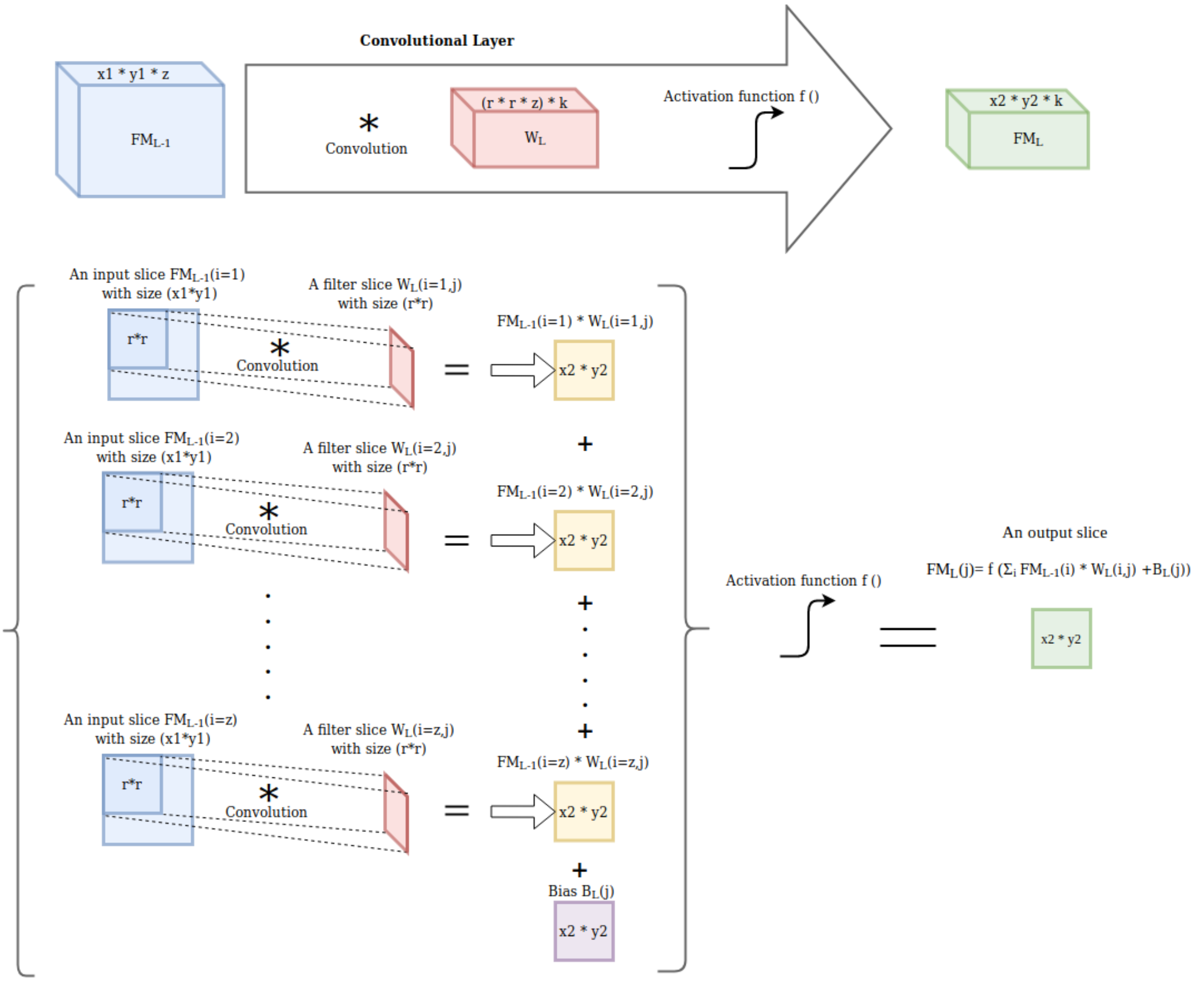

Convolutional Layer
x1 * y1 * z
FM$_{L-1}$
Convolution
(r * r * z) * k
W$_L$
Activation function f ()
x2 * y2 * k
FM$_L$
An input slice FM$_{L-1}$(i=1) with size (x1*y1)
r*r
Convolution
A filter slice W$_L$(i=1,j) with size (r*r)
FM$_{L-1}$(i=1) * W$_L$(i=1,j)
x2 * y2
+
An input slice FM$_{L-1}$(i=2) with size (x1*y1)
r*r
Convolution
A filter slice W$_L$(i=2,j) with size (r*r)
FM$_{L-1}$(i=2) * W$_L$(i=2,j)
x2 * y2
+
+
An input slice FM$_{L-1}$(i=z) with size (x1*y1)
r*r
Convolution
A filter slice W$_L$(i=z,j) with size (r*r)
FM$_{L-1}$(i=z) * W$_L$(i=z,j)
x2 * y2
+
Bias B$_L$(j)
x2 * y2
Activation function f ()
An output slice
FM$_L$(j)= f (Σ_i FM$_{L-1}$(i) * W$_L$(i,j) +B$_L$(j))
x2 * y2

Figure 1.2: A convolutional layer illustration for an input slice $FM_{L-1}(i)$ into an output slice $FM_L(j)$ using one filter $W_L(j)$.

As mentioned before, each convolutional layer acts as a dimentionality reduction for an input feature map FM_{L-1} with size $(x1 \times y1 \times z)$, to generate the output feature map FM_L of size $(x2 \times y2 \times k)$ computed as follow:

$$x2 = 1 + \frac{(x1 - r + 2 \times p)}{s} \quad \text{and} \quad y2 = 1 + \frac{(y1 - r + 2 \times p)}{s} \tag{1.1.3}$$

Where FM_L depth k is the number of filters in the the filter bank W_L. And p with s are arbitrary chosen hyper-parameters. In fact, p is zero-padding around the input border, and s is the stride at which we move each filter on input image during convolutions.

- **Pooling layer** this is also an important CNN layer. In fact, different pooling operation are available notably max pooling and average pooling, but they have all the same goal which is dimensionality reduction of an input tensor while preserving the spatial variance of his feature map in terms of translation and distortion [96,97]. For example, given an input feature map FM of dimension $(x1 \times y1 \times z)$ and an average pooling layer with a receptive field R of size $(r \times r)$ moving with a stride s. As presented in the equation below and figure 1.3, for each input feature map depth slice X of size $(x \times y)$, an element located at (p, q) and contained in the receptive field R is averaged to generate a new element located at (i, j) on the output depth slice Y with a reduced size $(x2 \times y2)$ [201], resulting in a final output feature map of size $(x2 \times y2 \times z)$ where:

$$Y(i,j) = Average(R_{p,q}(X)) \tag{1.1.4}$$

$$x2 = 1 + \frac{(x1 - r)}{s} \quad \text{and} \quad y2 = 1 + \frac{(y1 - r)}{s} \tag{1.1.5}$$

While depth z remains the same after the pooling operation.

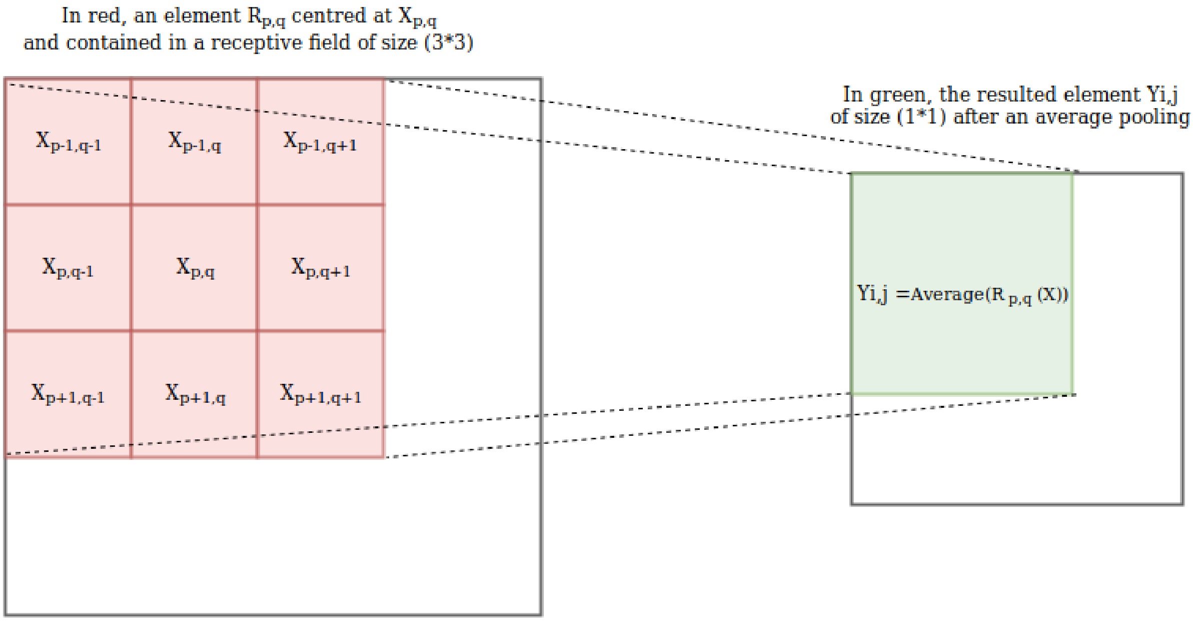

In red, an element $R_{p,q}$ centred at $X_{p,q}$ and contained in a receptive field of size (3*3)
$X_{p-1,q-1}$
$X_{p-1,q}$
$X_{p-1,q+1}$
$X_{p,q-1}$
$X_{p,q}$
$X_{p,q+1}$
$X_{p+1,q-1}$
$X_{p+1,q}$
$X_{p+1,q+1}$
In green, the resulted element Yi,j of size (1*1) after an average pooling
Yi,j =Average($R_{p,q}$ (X))

Figure 1.3: An average pooling illustration

- **Fully Connected** after stacking several convolution and pooling layers on each other, we usually add one or two Fully Connected (FC) layers at the end of this model. In fact, the first layers (convolution and pooling layers) aims to extract the common features between different classes, while these fully connected layers tries to understand their high level discriminative features [79,84]. A FC layer is fully connected to all activation map outputs in its previous layer. Therefore, its neurons map is calculated in a classic regression fashion as a matrix multiplication in addition to a bias offset. Generally, at the top of this pipeline we find a prediction layer. This layer could be the same as the last fully connected layer or another layer stacked on this last fully connected layer. For a given input image, this layer aims to compute its probability to belong to each class. Therefore, we use often a softmax probability distribution as a final predictor.

In order to learn and extract the most significant features, a CNN need to be trained. Training this model on a well defined problem with a fully labeled dataset such as BreakHis in a supervised manner is performed in an iterative manner. During each iteration, a batch of training images with their ground truth labels is provided to the network as input. After a feed-forward of this batch images through the CNN, the network computes in the last layer the error between actual outputs and the expected ones (ground truth labels). After computing the loss function, all layers filter banks weights and bias are updated in order to minimize the resulted error and fit the outputs with desired correct labels. This wights tuning operation is performed using backpropagation algorithm [96] where the error function gradient is propagated in the opposite direction through the network to adjust filter banks in order to minimize the output error that is affecting the model performance. After a number of feed-forward and backpropagation iterations, we could test the model classification performance on unseen data using test set images.

Now that most important CNN components have been presented, a CNN model for breast cancer images classification into benign or malignant tumors is illustrated in figure 1.4 below:

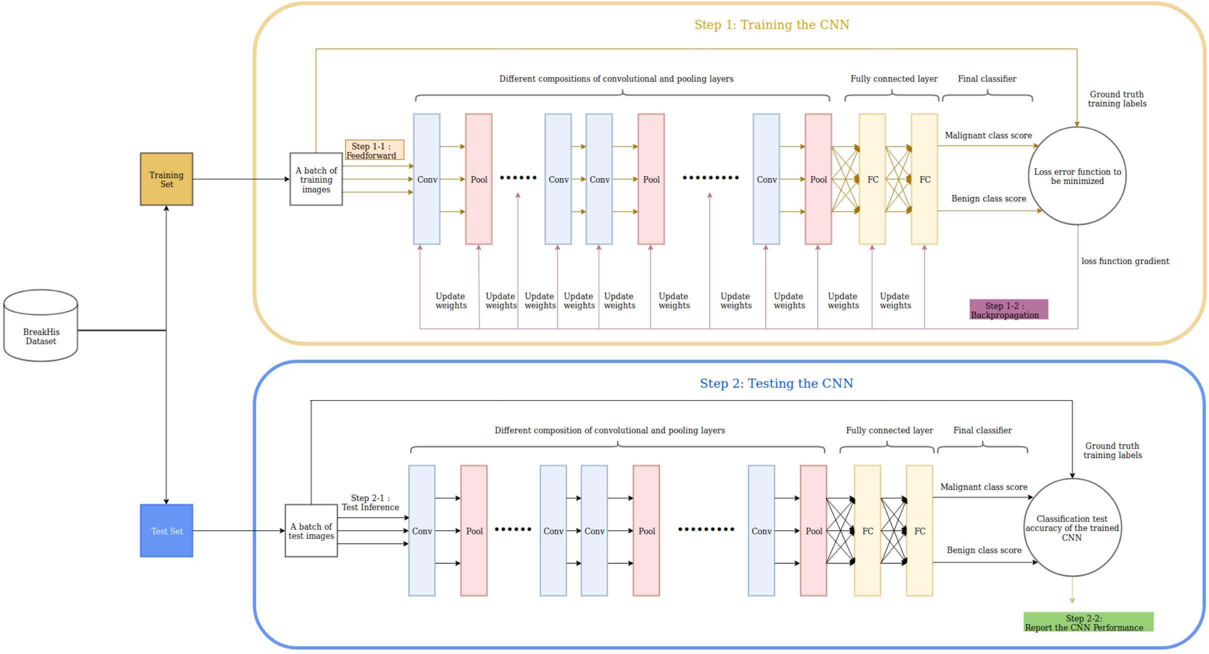


Figure 1.4: An illustration of a standard CNN trained for a binary classification of breast cancer images

For a full comprehensive review on CNN history and different architectures we recommend the reader this review [143]. For instance, we will focus on most used CNN models in biomedical and satellite imagery classification, and present a summary of their characteristics and architectural components:

- **LeNet** this model was the first CNN to be trained on a large dataset [96,97]. It contained 5 layers and a total of 60 thousands trainable parameters. This CNN called also LeNet-5 was introduced simultaneously with its application in MNIST dataset. This application proved CNN efficiency in a complex task at that time, which is hand written digits classification in bank checks.

- **AlexNet** After LeNet performance, practitioners spent several research years on CNN's loss minimization improvement due to gradient decent poor local minima phenomena [95]. Moreover, computational resources that used to be available were not sufficient to handle the large amount of data and parameters required to train a deeper CNN, in addition to other theoretical limitations such as those related to activation functions. During this period, Support Vector Machine (SVM) have known a great expanse in large application areas [143]. Until 2012, when Krizhevsky et al. Introduced a new 7 layers CNN called AlexNet [93] with 60 million parameters. AlexNet as most of its successor was validated during the ILSVRC competition and achieved a winning top-5 test error rate of 15.3% on 3-channel input images of size $224 \times 224 \times 3$, compared to 26.2% achieved by the second-best entry. Besides the huge improvement brought by this CNN in this benchmark challenge, AlexNet was introduced with a very efficient GPU implementation for more time and memory optimizations. Furthermore, to avoid overfitting, AlexNet used a dropout operation [79] by eliminating the neurons that contribute very poorly in the CNN output. In addition to that, it was the first CNN to use Rectified Linear Units (ReLUs) [125] as an activation function defined with $f(x) = max(0, x)$, because of its non-saturating non-linearity.

- **GoogLeNet** After 2 years, Google announced GoogLeNet or what is also known as Inception v1 [176] containing 22 layers with 4 million parameters instead of 60 million of its predecessor AlexNet. Despite of this important computational cost reduction, GoogLeNet won ILSVRC 2014 challenge. In fact, it achieved a very high classification accuracy rates with a to-5 error rate of 6.67%, claimed to be very close to human performance. The key of this success, is inception module implementation, the latter is composed of different small convolutions allowing better performance with a reduced parameters number. In addition to that, this CNN used an average pooling at the network top instead of fully connected layers, which allowed its authors to eliminate a large amount of fully connected parameters, and to prove that these parameters are not important to achieve a high accuracy. But the core component of this network remains the proposed inception module. Basically, this module acts as a bag of multiple convolution filters with some pooling. Then, all these operations results are concatenated, which allows the model to take advantage of multi-level feature extraction. Particularly, it extracts simultaneously general (5×5), (3×3) in addition to local (1×1) features. Then, their outputs are concatenated. Furthermore, GoogleNet uses two additional softmax prediction layers at different levels in addition to the standard softmax at the network top. These two additional softmax aims to enhance the gradient value which tends to vanish during backpropagation though such a deep network. To illustrate the inception module, an example is presented in the figure 1.5 below:

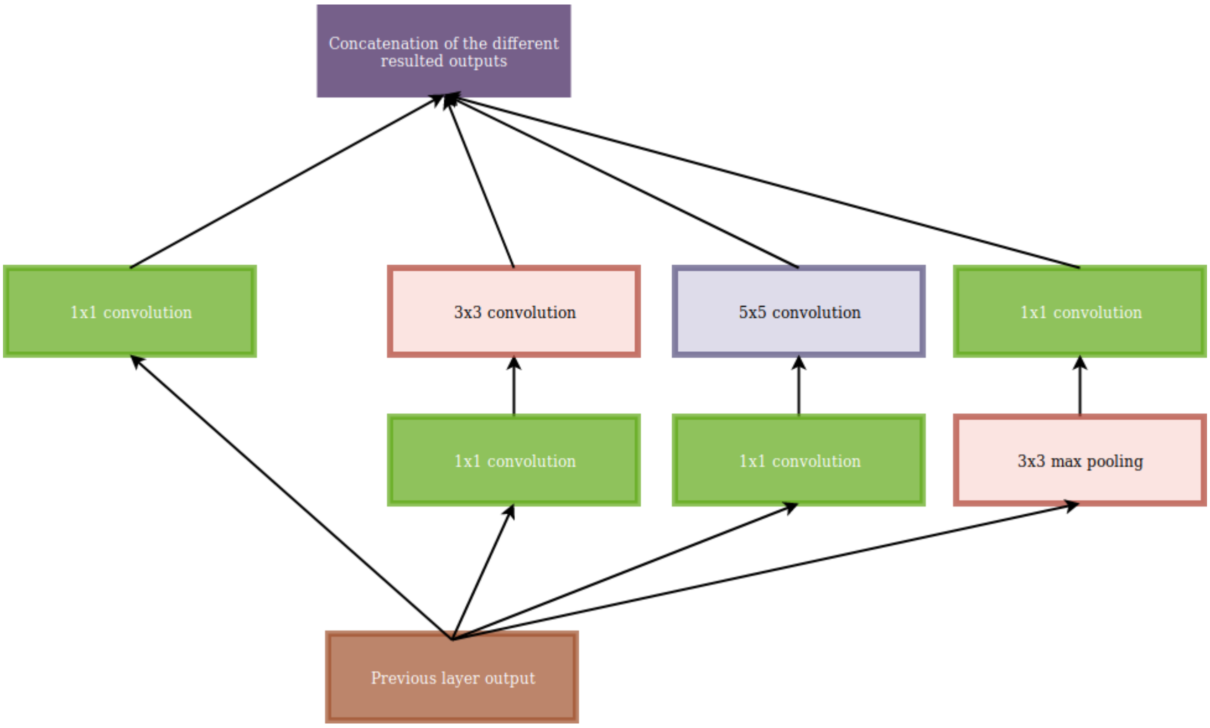


Figure 1.5: An illustration of inception module

Within its inception modules, GoogLeNet used small size 1×1 convolutions very intensively. This choice was highly inspired by the network in network architecture [103], with two main purpose. Firstly, they served as a dimensionality reduction to its prior computational-intensive convolution blocks $(3 \times 3$ and $5 \times 5)$. Secondly, they allowed the inclusion of rectified linear activation function RELUs [125]. Furthermore, GoogleNet CNN was improved many times resulting in several variants, but the most famous one is Inception-V3 [177]. In Inception-v3, authors adopted a new factorization of convolutions to gain more in computations. Additionally, Inception v3 achieved even higher performance when RMSProp [76] was adopted. Moreover, its authors proposed a new method to prevent overfitting called Label Smoothing, which is a regularizing component added to the loss formula to prevent the network from becoming too confident about a class at the expense of another. Furthermore, Inception-V3 used batch-normalization [83] at each output activation.

- **VGGNet** This network that will became known later as the VGGNet [161] was initially the runner-up in ILSVRC 2014 behind GoogleNet. Its best performing version in the challenge contained 16 layers. Its main contribution was in showing that the the network depth is a critical factor in achieving a better performance. Its proposed architecture was an extremely homogeneous pipeline that only performs 3×3 convolutions and 2×2 pooling in an end-to-end manner. However, a downside of VGGNet was being more expensive in computations and memory consuming, with a total number of around 140 million parameters, where most of these parameters were placed in the fully connected layers. At present, it was found that these FC layers can be removed with no performance downgrade, which significantly reduced the parameters number. Furthermore, similarly to GoogLeNet, different VGG variants were proposed throughout past years.

- **ResNet** After the infatuation of DL community on the development of more and more deeper CNNs. It was found that when these networks goes deeper, they becomes highly exposed to gradient vanishing issue and other optimization degradation problems. To overcome this depth limitation, ResNet authors [70] proposed what they called a residual function $F(x) = H(x) - x$, where $H(x)$ is the standard mapping function that we want to learn with an input x through few stacked non-linear layers (plain module). By reformulating it as $H(x) = F(x) + x$, where $F(x)$ and x represents the stacked non-linear layers and the identity function (identity shortcut connection) respectively. Based on their hypothesis, it is easier to optimize the reformulated residual mapping function $F(x)$ than optimizing the original mapping $H(x)$. This reformulation has been motivated by the counter-intuitive phenomena of degradation problem [70], where they observed that surprisingly during deeper CNNs training, we cannot fit the training data as we do in shallower CNNs training. By consequence, they claimed that the remaining deeper CNN's knowledge is hidden in this residual formulation. Their proposed ResNet based on residual module achieved outstanding results even with extremely increased depth (over 100 layers). In fact, RseNet outperformed all other CNNs in ILSVRC 2015 challenge with a to-5 error rate of 3.57%. Afterwards, many ResNet depth variants were proposed with different depths, but the most known ones are ResNet-50, RestNet-101 and RestNet-152. For better illustration, a residual module with the mapping reformulation in comparison to the standard one (plain module) is presented below in figure 1.6.

- **DenseNet** Lately, DenseNet was presented in [82] to take advantage from previous findings regarding CNN's depth increasing and shortcut connections. The specificity of this new network architecture is that each layer is connected to all its previous and next layers. In other words, each layer in this CNN is provided with feature maps from all its previous layers, while its own feature map output is provided to all its successors. To omit This fully connected fashion from leading to an enormous parameters number, its growth complexity is controlled by a regularization hyper-parameter k. In fact, this new proposed module connection called Dense blocks was motivated by its high density features propagation through the network and its features reuse at different layers. Inspired by ResNet [70, 71], DensNet authors introduced between each Dense block a composite concatenation function $H(\mathring{u})$ of three consecutive operations, firstly a Batch Normalization (BN) [83], followed by a rectified linear unit (ReLU) [125] and a 3×3 convolution. The main advantage of DenseNets is their improved features and gradients flow

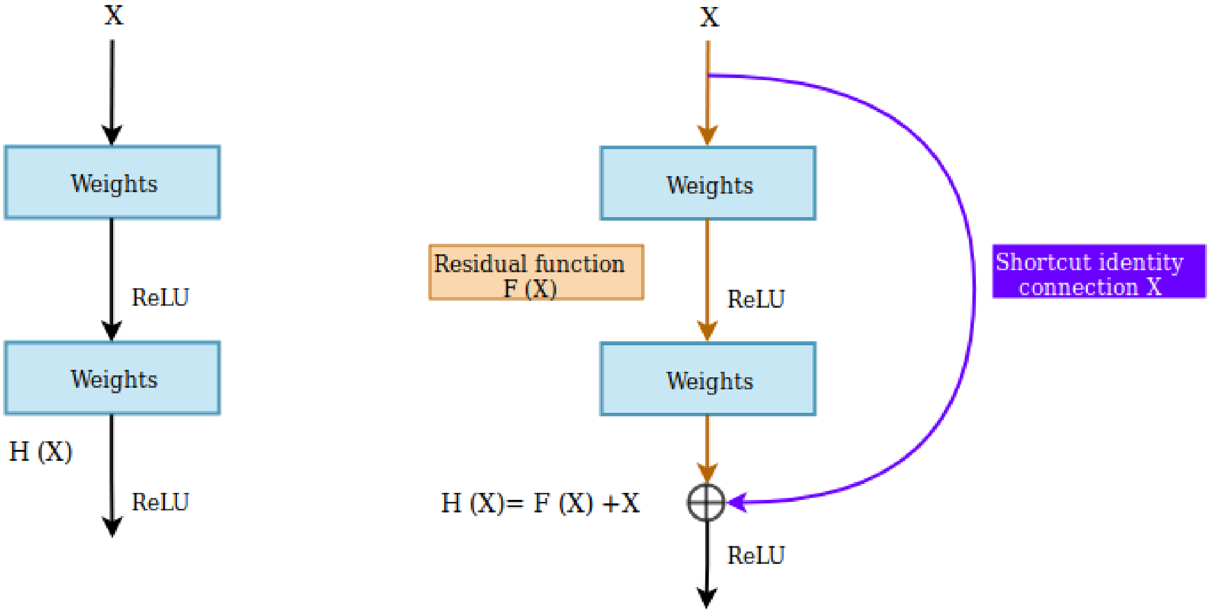


Figure 1.6: A comparison between a plain module (left) and a residual module (right)

throughout the network, making them more training efficient than their predecessors. This fact is mainly due to their connection strategy allowing each layer to have a direct access to loss function gradient and the original input signal, leading to an implicit deep supervision [99]. In fact, each DenseNet individual layer receives additional supervision from loss function throughout the shortcut connections, which is inspired by previously shown efficiency of deep supervision in Deeply Supervised Nets (DSNs) [99] which have their classifiers attached to every hidden layer forcing intermediate layers to learn discriminative features. In figure 1.7, we present a Dense block illustration.

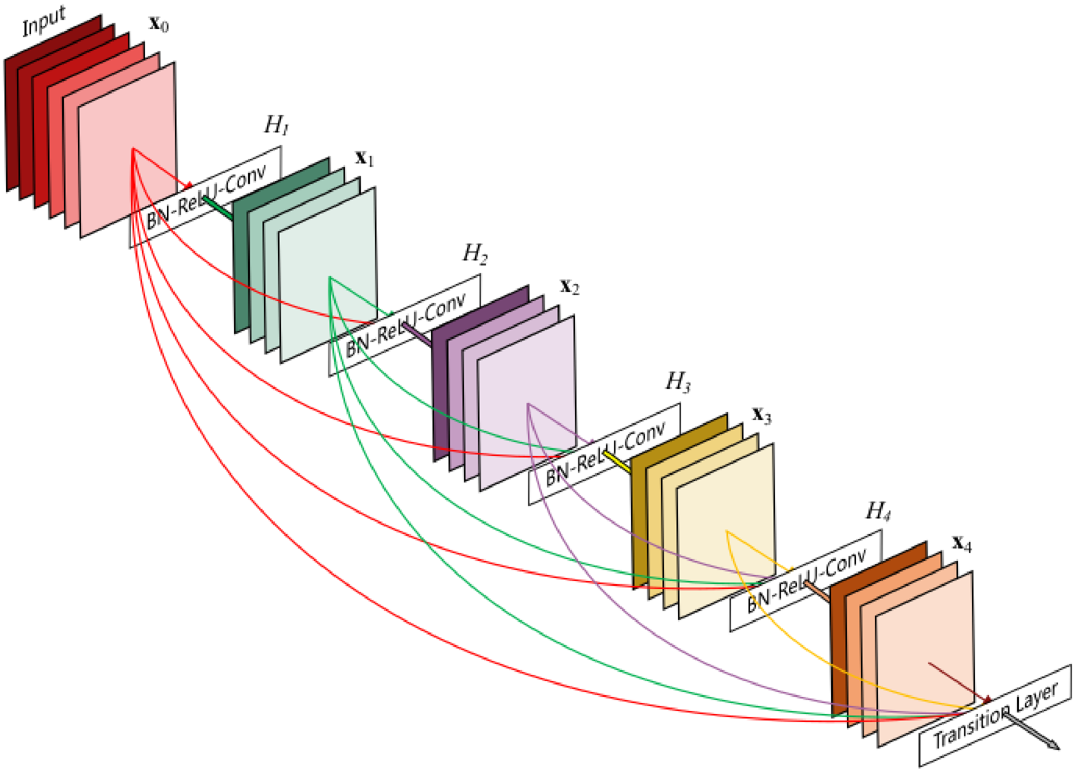


Figure 1.7: An illustration of Dense connections architecture taken from [82]

1.1.2.2 Generative Adversarial Networks (GANs)

Another type of DL models is Generative Adversarial Network (GAN). The latter is composed of two stacked networks where the first one is called the generator and the second one is the discriminator. The main idea is to have these two separated neural networks (generator and discriminator) locked in a competition with each other (this is where the name “adversarial” came from). The generator creates new images as similar as possible to original ones in a given dataset, while the discriminator tries to understand if they are original pictures or false ones (if they belongs to this dataset or not). In other words, the generator generates new data instances, while the discriminator evaluates them for authenticity (fake or real).
As presented in [58], we can think of a GAN as a competition between a paper money counterfeiter and a cop in a mini-max game, where the counterfeiter (the generator) is learning to pass false paper money, while the cop (the discriminator) is learning to detect them. The cop and the counterfeiter are both dynamic and in continuous training, and each one of them tries to learn the other’s methods. After this training, we end up having a generator that learned enough knowledge to sneak out the discriminator and generate new artificial samples that this discriminator will not be able to distinguish them as fake ones. An example of basic GAN with two stacked neural networks (a generator followed by a discriminator) applied to generate artificial new images as close as possible to the original ones, is presented in figure 1.8.

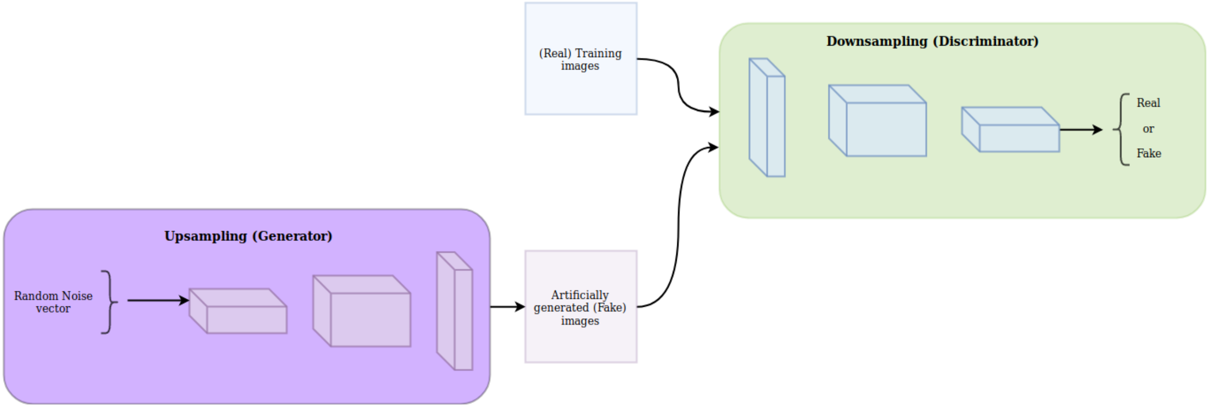


Figure 1.8: An illustration of a typical Generative adversarial network

1.1.2.3 Autoencoders (AEs)

In opposition to previous supervised learning networks, autoencoders acts in an unsupervised manner trying to learn features distribution of a given dataset. It was first introduced in one of the main unsupervised learning paradigms [148] to address the problem of unsupervised backpropagation, by using the input data as the only learning guidance. With the recent improvements in DL, autoencoders have taken center stage in different application areas, notably being used as a dimensionality reduction network [78]. In fact, the aim of such a dimensionality reduction autoencoder is to learn a mapping function $M_{W,b}(x) = x' \approx x$ in an end-to-end fashion throughout different stacked hidden layers mapping an input data x to its similar identity x' as depicted in figure 1.9 below. Generally, an autoencoder is composed of an encoder and a decoder. The first one is trying to learn a set of low dimensional representation features z while the second is trying to reconstruct a similar copy of the input data using only these learned intermediate features z. The identity function is a particularly trivial function to be learned; but often some constraints are placed on these hidden units, especially when z is having a lower dimension than x, and training the whole network become pushing these hidden units to learn the most representative features of input data.

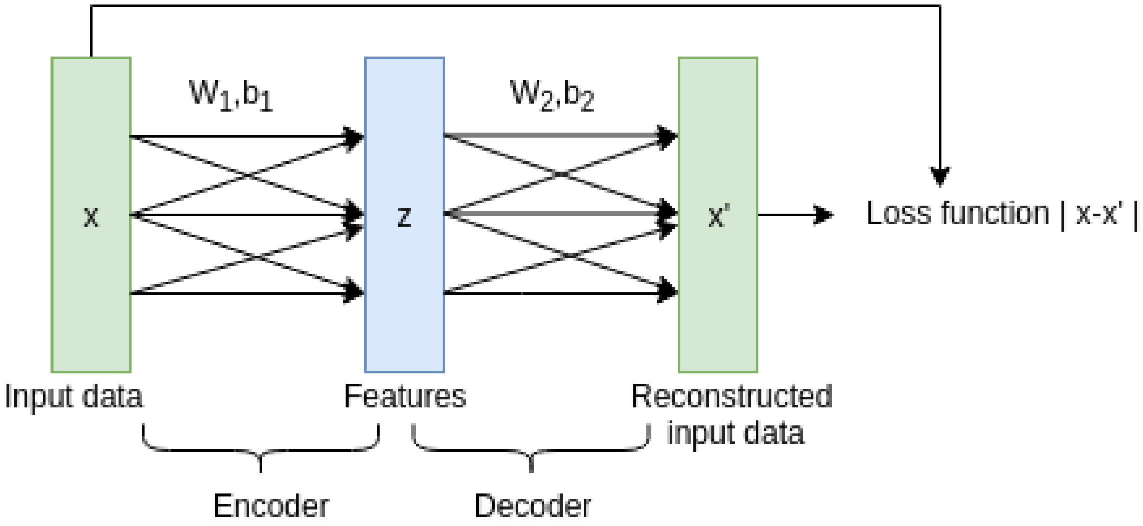


Figure 1.9: An illustration of a typical Autoencoder architecture

A special case of autoencoders, is called Sparse Autoencoder (SAE) [130], where sparsity is introduced into the hidden units by making the number of nodes in the hidden layer z bigger than that of the input layer x. In addition to that, when several (SAE) with only their encoding parts are stacked on each other, we obtain a Stacked Sparse Autoencoder (SSAE) which is often trained in a bottom-up greedy fashion. In fact, a (SSAE) model is able to learn deep feature representation from the data throughout its low-level (SAEs) until its high-level (SAEs) [130].

1.1.2.4 Deep Belief Networks (DBNs)

To learn deep features representation a Deep Belief Network (DBN) [77] is built with a concatenation of Restricted Boltzmann Machines (RBMs) stacked on each other. The core component of DBNs models [47] is the generative stochastic model called RBM that can be used either for unsupervised or supervised learning. It is composed of two layers, an input visible layer and a adjacent hidden layer trained with the aim to learn a probability distribution in the input set. Unlike original Boltzmann Machine (BM) [1], intra-connections between hidden-hidden or visible-visible layers in a RBM are disjointed forming a bipartite graph as illustrated in the figure 1.10.

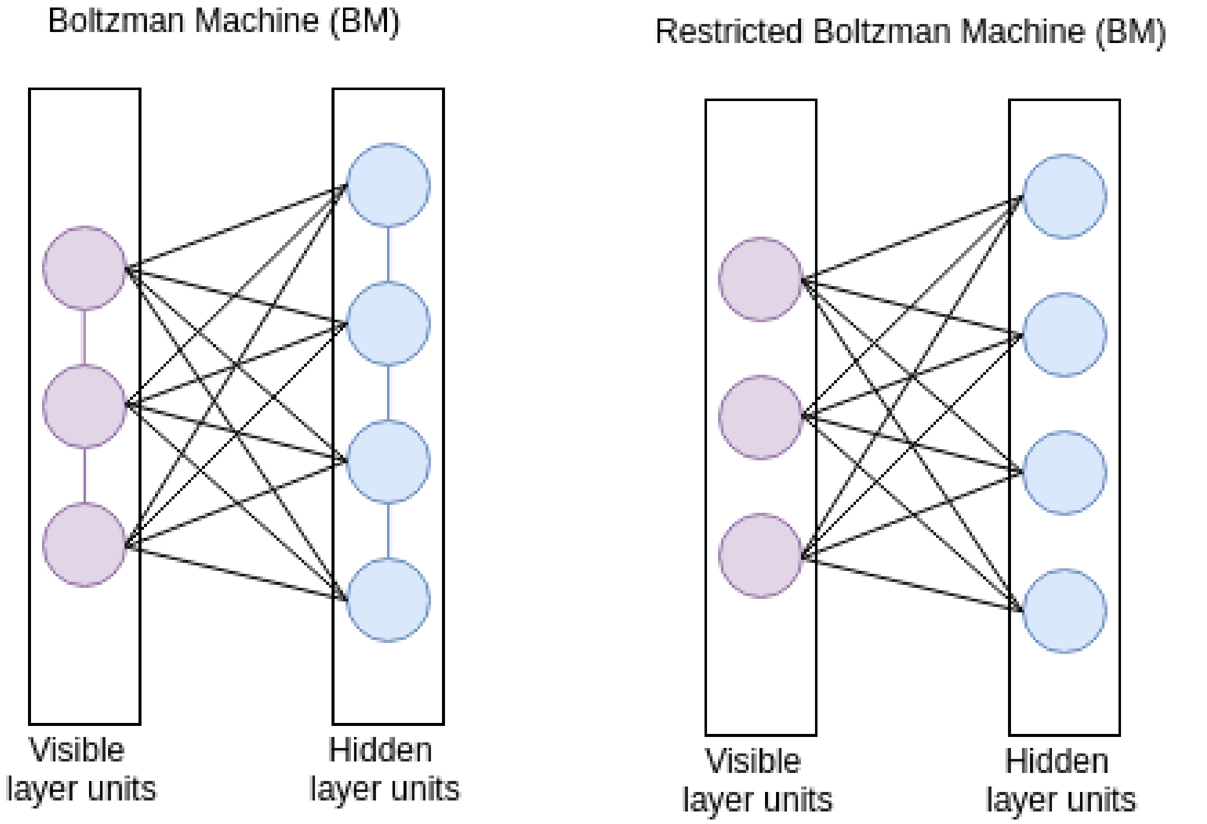


Figure 1.10: An illustration of the difference between a BM architecture(left) and RBM architecture (right)

1.2 Motivations

1.2.1 Deep Learning for biomedical images classification

Computer-aided diagnosis (CAD) has become a major research line for bioinformatics practitioners during the past few decades. Since the appearance of ML for images classification and the appealing performance of these models in this task, experts started to buid ML based CAD systems for the automatic diagnosis of many diseases. In fact, ML methods are trained on biomedical images collected from patients in hospitals and then used to makes assessment of the patient's condition by assisting clinicians in their decision-making process.

Automatic breast cancer diagnosis is one of the most important CAD applications. It involves different CV tasks such as tumor classification, localization and segmentation based on biomedical imagery. To elaborate these operations, CAD systems tries to learn the most significant features inside digitized histopathological images using annotated datasets containing this kind of images and ML classification models. To build CAD systems for breast cancer diagnosis, many datasets has been proposed. Most of these datasets has focused on tumor classification purpose only, since data annotation for tissue components segmentation (i.e. nuclei, cytoplasm and others) or tumors localization, is a very tedious, expensive and time-consuming task. In this thesis, we focus on one of the most recent benchmark datasets for tumor classification called BreakHis. The latter contains annotated histopathological images collected from both malignant and benign patients. Moreover, BreakHis offers a more detailed annotation of tumors malignancy into four sub-types. Hence, when used in conjunction with ML classifiers, it allows a binary classification of slides malignancy (benign or malignant tumors), as well as the ability to classify each one of these two main classes into four different sub-categories depending on breast lesion appearance.

Traditionally, ML image classification models for this kind of CAD systems are built in a dual-step process [143]. The first phase consists in a handcrafted features extraction task using various features descriptors. In the second step, these extracted features are further used to train a standalone classifier to map each image to its corresponding malignancy class. A large number of CAD systems have been built with this classical approach, including those trained with BreakHis dataset. However, this approach presents a major drawback, which is: the classifier's accuracy relies essentially on the prior extracted features, whereas getting high quality features in such a complex problem remains a very difficult task [98].

Lately, many DL models have been proposed to overcome these limitations. The key advantage of these models is that they are able to automatically elaborate the features extraction and classification steps into one unique black box mapping function through an trainable ANN. To date, the most efficient DL models for these kind of CV tasks are CNNs [95]. Indeed, all significant works on BreakHis dataset adopted these architectures as a base model. However, the main problem related to these models remains the training strategies and the data quality used to train them as we introduced in the thesis abstract. In fact, the intrinsic complexity of BreakHis dataset imposes on the user to take into consideration its data quality, classes imbalance and the lack of sufficient samples when training DL models. Therefore, researchers often performed pre- and post-processing methods to increase BreakHis data value and take the most out of it when using DL. In the first part of this thesis, we elaborated an overview and a performance comparison of all available CAD systems based on this benchmark dataset. Then, using all lessons learnt from this analysis, we built an ideal breast cancer CAD system using DL with our proposed data pre- and post-processing techniques.

1.2.2 Deep Learning for satellite images classification

During the past years, there has been an extensive amount of remote sensing data growing and collected everyday by satellites orbiting around our planet, drones or other types of remote data collectors. Particularly, a large amounts of high resolution remote-sensing images at the global scale are acquired daily with a good quality and an accurate precision using very sophisticated satellites. Simultaneously, a notable growing in the popularity of ML methods and especially DL models in various remote-sensing applications has been reported. And thanks to these factors, many research fields are achieving important breakthroughs, such as geo-spatial object detection, LULC scene classification, weather forecasting and other applications.

The availability of this large amount of high quality data has played a big role in the development of remote sensing field in comparison to other applications such as CAD systems elaboration. In fact, having good quality data remains the most attracting and encouraging element for every person in the research community. In addition to that, we all know that the required expertise to handle biomedical images is much bigger than the one needed to analyse remote sensing imagery. A very good example and highly solicited research area in this discipline is LULC classification using remote sensing data and especially satellite imagery. LULC mapping is very important field in earth observation science and serves to establish a better understanding of our planet. Moreover, LULC mapping allows a better monitoring of natural resources, agriculture and vegetation preservation, helps a better decision making for urban planning and prevent natural disasters.

As in the biomedical imagery classification, in remote sensing, classical ML methods needs a prior feature selection phase to further classify the various LULC types present on the images. Thus, DL models and especially CNNs that are able to automatically extract and map the most important visual features to each LULC class seems to be the best choice in this kind of applications. However, satellite images classification using DL can be a very complex task especially with images collected from different regions of the planet and under different conditions as it is the case for all available LULC datasets. In fact, many datasets with annotated satellite images has been proposed with the aim to explore their potential using CNNs. Nevertheless, the sufficient and optimum accuracy degree has not been reached yet. This is mainly due to the factors exposed above and fact that most datasets suffers from a lack either at the image quality level, the spatial resolution, the annotation quality, their size or their geographical coverage. All these drawbacks affect the DL model quality and by consequence its LULC mapping capability. To circumvent these limitations, we proposed Sentinel2LULC, the first global, high resolution and freely available satellite images dataset for LULC mapping. During the building of this dataset, we took into consideration all the necessary requirements to train CNN models and made it ready-to-use for this specific purpose.

Creating our own dataset for this very complex task has allowed us to further analyse DL performance under different conditions. And in this new scenario, the preprocessing that we have elaborated was during the creation of the dataset itself. In this new setting we have a complete control on the data quality that we will use to train DL models. All DL experiments and evaluation reported for this second use case of the thesis were carried out using Sentinel2LULC dataset in conjunction with the most popular CNN models. And an analysis within these new circumstances has offered us a way to evaluate either the data quality is the only and sufficient factor that was lacking in the first use case application to reach a good performance or not.

1.3 Contributions and outline of the thesis

In this thesis, we propose adequate and efficient preprocessing techniques for two DL applications. In the first application, we applied these preprocessing techniques in conjunction with DL models on an already available dataset for automatic breast cancer diagnosis. Whereas, in the second use case, our preprocessing methods were established during the dataset creation itself and allowed us to ensure that the data quality is high enough to be used with DL in order to build an automatic LULC mapping system. This analysis has also allowed us to evaluate the importance of data quality in DL training for these two complex applications. The rest of the thesis is organized as follow:

In Chapter 2 of this thesis, which constitutes the first part of the published paper in Neurocomputing Q1 journal, we will introduce the first complex application: DL models for biomedical images classification. First, we will present BreakHis dataset, then, we will give an overview on all models and methods used with this dataset to build automatic breast cancer CAD systems, and given the prevalence of DL models in our study, we will highlight the achieved performance in all BreakHis based works who explored DL models. This overview will be also organized given different adopted pre- and post-processing approaches including DA methods, TL approaches and training settings.

In Chapter 3 which constitutes the second part of the published paper in Neurocomputing Q1 journal and the entire paper published in the Proceeding of LOPAL international conference, we used all learnt lessons from the previous section to establish what would be the ideal CAD system for BreakHis based CAD system from a practical as well as a clinical standpoint. This CAD system is built from the most adequate DL model, pre- and post-processing methods for this problem. Then, we implemented and evaluated its performance under different pre- and post processing settings.

In Chapter 4 submitted to Scientific Data Q1 journal, we will start tackling the second complex use case of this thesis which satellite images classification using DL for LULC mapping. In this chapter, we introduce Sentinel2LULC a Sentinel-2 RGB imagery dataset that we have carefully created to be specifically used in conjunction with DL in order to build automatic global LULC mapping.

In Chapter 5 to be submitted to a Q1 journal, we explored the performance of different CNN models in conjunction with Sentinel2LULC for global LULC mapping. Then, we explored geographically and in details the best performing CNN in order to find the correlation between the geographic distribution of the data used in this study and the classification performance.

Finally, Chapter 6 gives the conclusions of this thesis and suggestions for the future works to be elaborated.

CHAPTER 2

An overview on data preprocessing and deep learning post-processing methods for BreakHis based breast cancer computer-aided diagnosis

Contents

This chapter constitutes the first part of the paper published in Neurocomputing [15].

2.1 Introduction and motivation

Let us first give a detailed introduction to the main topic of the present chapter: In spite of the massive growth in breast cancer incidence during last years, its death rate has considerably decreased [46]. This drop in mortality incidence has mainly occurred in developed countries who achieved important breakthroughs in early detection methods through medical imaging analysis [45]. The most infallible early breast cancer diagnosis method in clinical routine is biopsy examination [192]. The latter is carried out by pathologists using fine needle expelled slides from breast tissue. For each patient, an important number of breast tissue slides are analyzed with various microscopic magnification levels to better highlight Regions of Interest (ROI). Nevertheless, pathologist's interpretation at decision moment could be deviated by several human factors such as eye fatigue, in addition to instrumental-dependant factors, notably those related to the used microscopic device.

To alleviate the risk of putting a patient life at stake, domain experts thought of entrusting their assistance in this difficult task to Computer Aided Diagnosis (CAD) systems [6,55,66]. In addition to breast cancer, many high risk diseases are now diagnosed using these AI solutions [81], and a large research community is continually trying to improve its diagnosis efficiency. The main objective of these researches is to make these CAD systems able to help domain experts making the most accurate decision to the departure of their patients. To achieve a good performance during diagnosis support, the golden standard knowledge source for these systems is data collected by experts from real decision-making situations. For this reason, many breast cancer diagnosis datasets are proposed [5,53,85], but their main limitation remains that most of them are not publicly available to research community. Moreover, even when this availability issue is overcome, often public datasets suffers from a lack in sufficient clinical value to build a reliable CAD system. To our knowledge, the most representative breast cancer dataset able to overcome both limitations is BreakHis [169].

BreakHis is a recent breast cancer public dataset. Its clinical potential consists of its two-level annotation, which gives information about the malignancy and the exact tumor category for each biopsy slide. In fact, the first annotation level allows a binary classification according to slides malignancy (i.e benign or malignant tumors), while the second level enables a multi-category classification option to further classify each one of both malignancy classes into four different sub-categories each depending on breast lesion appearance in these slides. Additionally, BreakHis images were acquired with four microscopic magnification levels ($\times 40$, $\times 100$, $\times 200$, $\times 400$), which allows practitioners to either train a specific model for each magnification level subset (training with a magnification-specific approach) or train a unique model with all magnification level images combined (training with a magnification-independent approach).

Since its release in 2016, over 40 studies analyzed BreakHis potential in building breast cancer CAD systems. Authors of these studies reformulated this problem as either one of the following:

- A magnification-specific binary classification
- A magnification-independent binary classification
- A magnification-specific multi-category classification
- A magnification-independent multi-category classification

To address each one of these classification tasks, researchers used various pre- and post-processing methods with different learning models and continues to achieve more accurate results, especially with the recent arrival of outstanding DL models such as CNNs, known for their human-close performance in several image classification problems.

To summarize all these research, the present chapter gives a concise overview of all BreakHis based CAD systems. Along this overview, we highlight more the DL based CAD systems and establish an analytical comparison of their used pre- and post-processing approaches.

2.2 BreakHis histopathological breast cancer dataset

This section provides a complete description of BreakHis dataset, the experimental protocol established by its authors, in addition to a discussion of its limitations.

2.2.1 BreakHis dataset description

As is the case for most cellular pathologies, histopathological slides are the core element for breast tumors examination [6]. Histopathological slides are surgically expelled tissues from a given patient after a biopsy operation on the area of interest. BreakHis current version is composed of 7909 histopathological biopsy images taken from 82 patients. These images were collected by P&D Laboratory in Brazil from January 2014 to December 2014. BreakHis is divided into two main malignancy classes: benign and malignant, with 2480 benign and 5429 malignant tumor images. Each malignancy class is distributed into four different sub-categories based on the tumor appearance under the microscope. Benign breast tumors are divided into the following sub-categories: Adenosis (A), Fibroadenoma (F), Phyllodes Tumor (PT), and Tubular Adenoma (TA). Malignant tissues are divided into four sub-categories: Ductal carcinoma (DC), Lobular Carcinoma (LC), Mucinous Carcinoma (MC), and Papillary Carcinoma (PC). Each patient in this dataset has a different number of collected images which are annotated with their main malignancy class and corresponding subcategory. A summary of image and patient distributions over main classes and different sub-categories is presented in Table 2.1.

Main category	**Benign**				**Total**	**Malignant**				**Total**	**Total**
Sub-category	**A**	**F**	**TA**	**PT**	**Benign**	**DC**	**LC**	**MC**	**PC**	**Malignant**	**both**
Number of images	444	1014	453	569	2480	3451	626	792	560	5429	7909
Number of patients	4	10	3	7	24	38	5	9	6	58	82

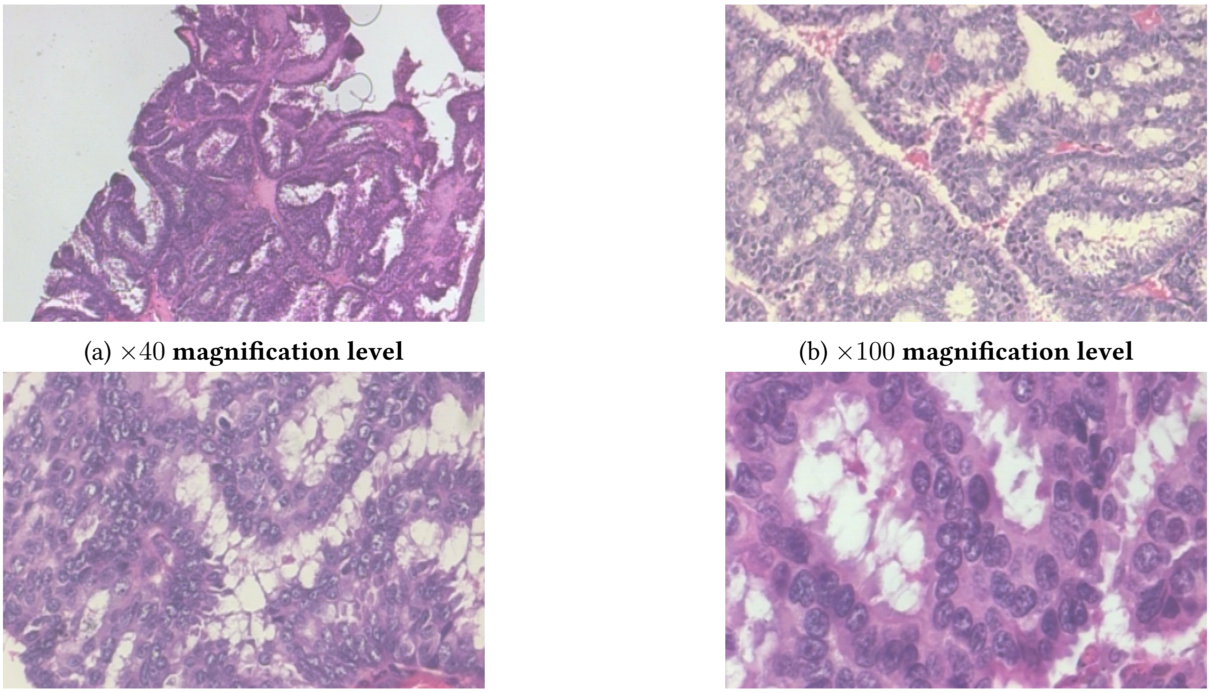
Table 2.1: Image and Patient distribution among the main categories and each sub-category.

BreakHis images were collected using the same clinical process adopted in similar histopathological datasets [191]. For each patient, several breast tissue samples were aspired with a fine biopsy needle in the operating room. Then, each sample undergoes the following preparation phases:

- Starting with formalin fixation and embedding in paraffin to preserve the original tissue structure and molecular composition
- Then, sections with $3\mu m$ of thickness were extracted from paraffin outcomes using a high precision cutting instrument called microtome
- Subsequently, these sections were mounted on covered glass slides for visualization under microscope

Generally, components of interest such as nuclei or cytoplasm are not clearly visible inside raw tissue on mounted sections. Thus, an essential operation called tissue staining takes place before visualization. This staining step aims to highlight each morphological component separately for a better visual insight under microscope. In fact, several staining methods exists, and the most used one is Hematoxylin and Eosin (H&E) [6] as used for BreakHis. In H&E, hematoxylin binds to DNA and thereby dyes the related structures of interest (i.e. in most cases nuclei) with blue/purple, and eosin binds to proteins and dyes other structures including cytoplasm and stroma with pink.

Several years ago, this workflow was concluded by sending to pathologists all stained-slides in physical version for analysis and annotation. Nowadays, with the appearance of Whole Slide Image (WSI) scanners in these pathology labs, a slide digitization step is added on the top of this workflow, and a digitized version of these slides is also sent to pathologists, then annotated and stored in the laboratory information system. Furthermore, this additional step may includes slide manipulation and loading operations, which allows laboratory to collect slides with different magnification factors as done in BreakHis, where images were taken with four magnification levels (×40, ×100, ×200, ×400). During analysis and annotation, pathologists starts by identifying ROI in the lowest magnification level slide (×40), then dive deeper in this ROI with higher magnification levels (×100, ×200) until reaching a bigger insight on this region (×400). To illustrate this process, a BreakHis slide example captured with four different magnification factors is presented in figure 2.1.



(a) ×40 **magnification level**

(b) ×100 **magnification level**

(c) ×200 **magnification level**

(d) ×400 **magnification level**

Figure 2.1: The same malignant PC(Papillary Carcinoma) tissue captured with different magnification factors, taken from the patient with ID:9146 in BreakHis dataset

BreakHis images distribution into these four magnification levels for each tumor category and subcategory is presented in Table 2.2. In fact, BreakHis images are almost equally distributed between different magnification factors with a narrow range starting from 1794 images in ×400 subset to 2051 images in ×100 subset. For BreakHis patient distribution, each magnification factor subset contains exactly 82 patients because each patient has images taken with all magnifications. After further statistical exploration, we found that each magnification factor subset contains around 24 images per patient. In average 24, 25, 24, and 22 images per patient are available in ×40, ×100, ×200, and ×400 subsets respectively.

Main category		Benign				Total	Malignant				Total	Total
Sub-category		A	F	TA	PT	Benign	DC	LC	MC	PC	Malignant	of both
Number of images at each magnification level	×40	114	253	109	149	598	864	156	205	145	1370	1968
	×100	113	260	121	150	614	903	170	222	142	1437	2051
	×200	111	264	108	140	594	896	163	196	135	1390	1984
	×400	106	237	115	130	562	788	137	169	138	1232	1794

Table 2.2: The distribution of BreakHis images into four magnification levels for both main tumor categories and each sub-category.

2.2.2 BreakHis data imbalance and noise

BreakHis histopathological dataset has been released to overcome the lack of previous datasets in terms of availability and clinical content richness. Nevertheless, BreakHis also suffers from other common issues that are frequently present in all medical datasets due to the disease nature and limitations in medical data acquisition. The following limitations are the most relevant in BreakHis dataset, and they should be taken into consideration when building a robust CAD system:

- **Data imbalance** As it can be seen in Table 2.1, BreakHis data imbalance occurs at different levels:
  - The uneven patient distribution between main malignancy categories (malignant and benign) with an Imbalance Ratio (IR) of 0.41 (24 benign patient versus 58 malignant patient).
  - The uneven image distribution between main malignancy categories (malignant and benign) with an Imbalance Ratio (IR) of 0.45 (2480 benign images versus 5429 malignant images)
  - An uneven distribution is also present between different sub-categories at image and patient levels. For example, the benign sub-category (F) has 1014 images captured from 10 patients, while a malignant sub-category like (DC) has 3451 images collected from 38 patients

  In fact, this data imbalance issue could bias the discriminative capability of a CAD system towards the majority malignant class at image and patient levels during binary and multi-category classification tasks. Therefore, one should be aware of this limitation when building any classification model.

- **Label noise** During its evolution, breast tumor expands gradually from a region to another in breast tissue. Hence, sometimes images captured from the same patient may contain tissue samples from different regions, and by consequence the same image could contains different breast cancer stages, which means different sub-category annotations for the same image. Notably, images taken from the malignant patient with ID:13412 which contains morphological features of two malignant sub-categories, ductal (DC) and lobular carcinoma (LC). This special case can be considered as a noisy labeled patient, and able to confuse the CAD model during its training on multi-category classification task that tries to learn discriminative features between both malignant sub-classes (DC and LC).

2.2.3 BreakHis experimentation protocol

BreakHis has been proposed with the aim to constitute a benchmark for breast cancer CAD systems. Thus, its authors proposed in [169] the following unified experimentation protocol:

- For each magnification factor subset, five evaluations are performed.
- During each evaluation the used magnification subset is randomly divided into 70% for training and 30% for test.
- To guarantee that the CAD model generalizes to unseen patients, the patients used to build the training set are not used for test.

For a fair comparison between different CAD systems, BreakHis authors proposed two classification level metrics:

- **Image Level Accuracy (ILA)** This first metric is the standard classification measurement at image level. It does not take into account the patient information and it is defined as follows:

  $$ILA = \frac{I_{corr}}{I_{tot}} \tag{2.2.1}$$

  Where I_{tot} is the total number of test images and I_{corr} is the number of correctly classified images by the evaluated model.

- **Patient Level Accuracy (PLA)** The second metric reflects the achieved performance in a patient-wise manner. Firstly, an individual score is computed for each single patient, and then the mean accuracy is calculated over all test patients.
  For a given patient P_i, its patient score $P_{score}(P_i)$ is:

  $$P_{score}(P_i) = \frac{I_{Pcorr}}{I_{Ptot}} \tag{2.2.2}$$

  Where I_{Pcorr} and I_{Ptot} are respectively the number of correctly classified images and the total number of images for this patient.
  Then the Patient level accuracy (PLA) is measured by:

  $$PLA = \frac{\Sigma_{i=1}^{i=N} P_{score}(P_i)}{N} \tag{2.2.3}$$

  Where N is the total number of patients in the test set.

- **Additional metrics** To guarantee a fairer comparison between different models regardless of the uneven data distribution between available classes, several related works adopted other evaluation metrics such as:

  - F1-score [162], also called F-measure or F-score, was adopted to better highlight the evaluated model's sensitivity to malignant cases which are of higher interest in this kind of medical diagnosis. Conventionally, a malignant case is considered as positive while a benign one is considered as negative, and F1-score is the harmonic mean between Recall and Precision where:

    $$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{2.2.4}$$

    $$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{2.2.5}$$

    And F1-score is calculated as follow:

    $$F1 - score = \frac{2 \times (Recall \times Precision)}{Recall + Precision} \tag{2.2.6}$$

  - Area Under Curve (AUC) [162] was adopted in other works on BreakHis dataset to evaluate their CAD systems accuracy. This metric was mainly used to illustrate the probability that a positive case outranks a negative one according to the used classifier.

2.3 Histopathological images preprocessing methods overview

Models used to build CAD systems relies essentially on the data quality provided to them. Therefore, usually original data undergoes several preprocessing steps before feeding these models. In this section we will provide a general definition of the most used preprocessing methods for histopathological images, especially those used in almost all BreakHis related works: stain normalization and data augmentation. In addition to some variants of these methods that have never been explored with BreakHis images yet.

2.3.1 Stain normalization

Alike pathologists, CAD systems relies considerably on histopathological images staining quality [80]. However, due to variations in device settings, experimental parameters, staining protocols, in addition to differences between slide scanners and patients tissues, often stained slides in the same histopathological dataset are inconsistent and prone to huge appearance variability [153]. As shown in figure 2.2, BreakHis has also a high variability degree in tissue appearance caused by its staining irregularity. These variations not only leads to an inconsistency between different pathologists verdicts, but also hinder the CAD systems learning process [31]. By consequence, it is important to consider a data preprocessing step to mitigate this staining variability before any color, texture, or stain based features extraction for breast cancer classification. A natural and straightforward manner to adress this issue would be a gray-scale transformation [12, 169], but the latter may cause an unwanted loss of very important characteristics and discriminative colors. Thus, the most used approach is stain normalization preprocessing which besides being more efficient, it allows color-based information preservation by transforming colors range instead of eliminating them completely. Practically, several color transformation methods have been adopted for stain normalization and they can be divided into three broad categories:

- **Color-matching methods** These methods normalizes all under or over stained histopathological images using a reference image. Indeed, their goal is to grant these images the same colors distribution as in the reference image. The most used method in this category is called Reinhard [144], it matches the color distribution of an image to that of the reference image by applying a linear transformation in a perceptual color-space so as to match each color channel mean and standard deviation for both images.

- **Stain-separation methods** These methods normalizes each staining channel (Hematoxilyn and Eosin) independently. The most used one in this category is Macenko [108]. It finds stain vectors using a reference image and a transformation from RGB space to Optical Density (OD) space. Then, normalizes each histopathological image by converting its color values to their corresponding optical density (OD) values.

  An illustrative stain normalization transformation of some BreakHis images using Reinhard and Macenko methods is shown in figure 2.2.

- **Deep learning based preprocessing methods** The main drawback of the two last categories is the need of a domain expert to pick a high-quality reference image from the used dataset, which is a hard decision to make. More importantly, it's clear that limiting this process to one image with very specific staining characteristics could lead to a poor generalization when applied to other images with new spatial structures and different morphological components. Nowadays, with the growth in generative capability of deep learning models especially of Generative Adversarial Networks (GANs) [58](these architectures are presented in Section **??**), a new learning-based

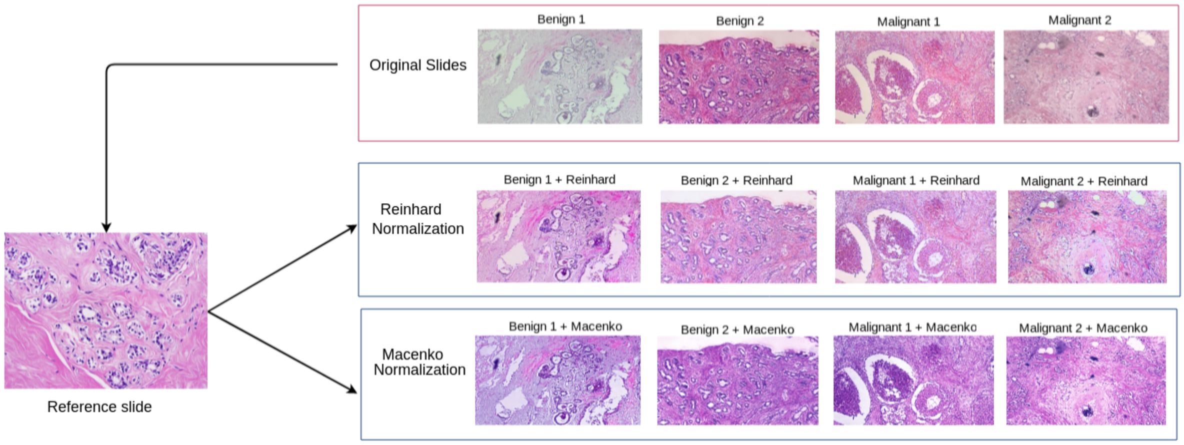
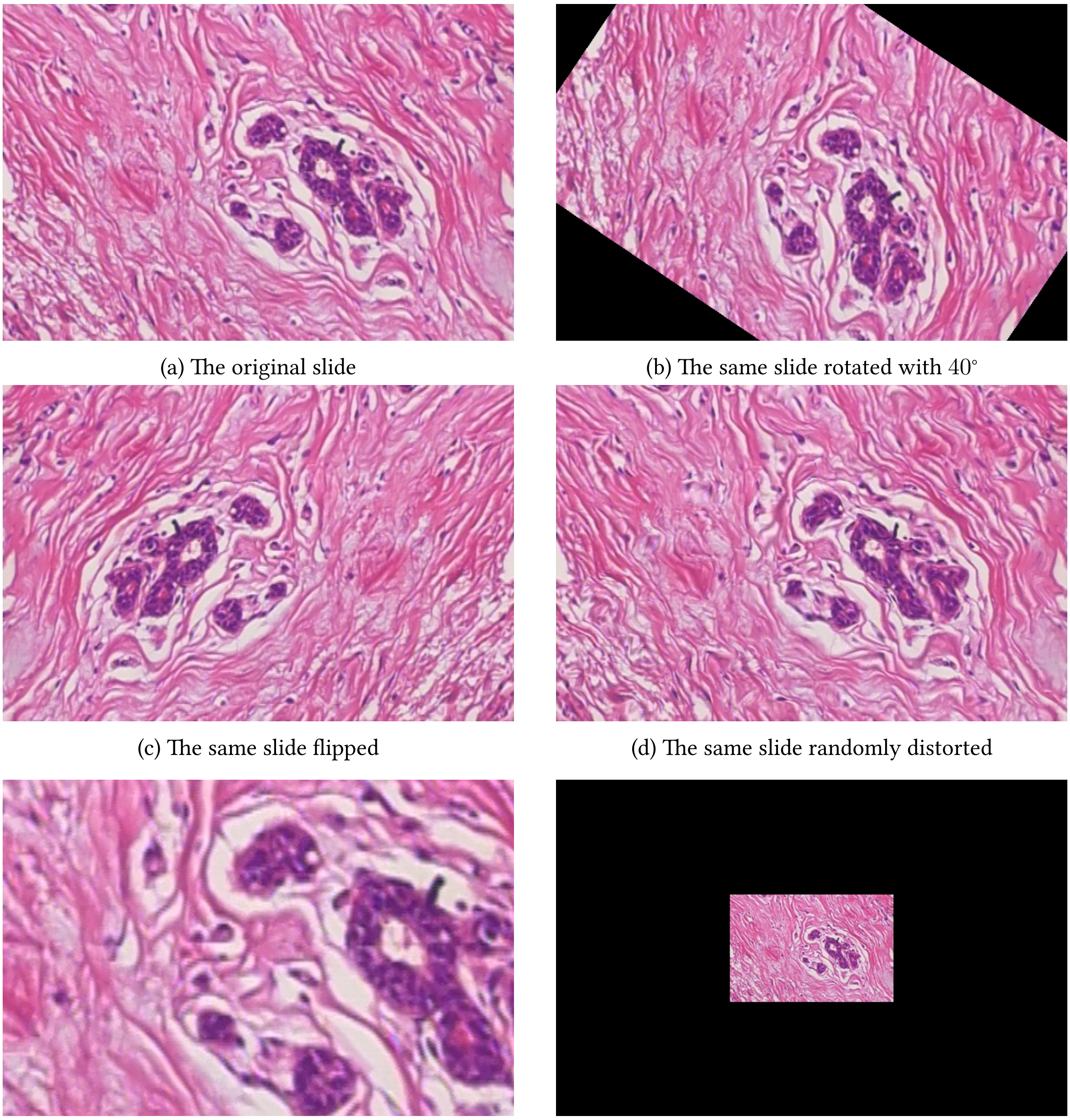


Figure 2.2: Stain normalization of four slides from BreakHis dataset. Two benign images (Benign 1, Benign 2) and two malignant images (Malignant 1, Malignant 2). The first row shows raw images before normalization, the second and third rows shows normalized images using Reinhard and Macenko methods respectively. Both normalization methods have been applied using a randomly chosen BreakHis reference image shown in the left corner of this figure.

category of stain normalization is emerging. In fact, some recent GAN-based stain normalization methods [18, 153] appears to be very efficient in achieving this task. In spite of that, GAN based methods have never been experimented yet on BreakHis images for stain normalization. The most attractive method among them is StainGAN [153], where the authors proposed an end-to-end Generative Adversarial Network (GAN) trained to transfer H&E staining colors between two different scanners datasets allowing a wider generalization margin. Moreover, this StainGAN model is trained to transfer staining style between different datasets without the need of paired (input-output) slides. In fact, this model is inspired from cycle-consistent GAN concept [213] allowing a weakly supervised staining mapping between images taken with two different scanners, this cycle-consistency is a constraint on distance preservation between a given input image and the reconstructed one (i.e. the normalized image) through the generative model. Hence, the staining appearance is transferred between different scanners, while textures and all other morphological components describing the input image are preserved in the generated one. We believe that these learning-based approaches are capable to overtake the classic stain normalization methods by giving for practitioners more opportunities to extract any normalized scanner staining style and transfer it to another dataset, while less relying on domain expert's availability and additional expenses.

2.3.2 Data augmentation

Generally, efficient machine learning models used for classification and particularly DL ones, have very large architectures with millions of learnable parameters. To achieve an acceptable generalization rate while avoiding overfitting, classification models requires a high training samples number [136]. However, obtaining large and high quality clinical labeled dataset is a very expensive and time consuming task. To overcome this limitation, practitioners proposed to transform available data to generate new artificial data samples. The most used approach for this purpose is data augmentation. In fact, after proving its ability to bring a huge improvement to deep learning models, data augmentation has rapidly attracted a great interest from research community, and continues to improve along with the growing popularity of CNNs [117]. Particularly, in many BreakHis related works, raw images were rotated, flipped, distorted, zoomed in or out to generate new images with the same discriminative features as the original ones. Data augmentation can be performed either offline by external pre-training image processing or online during the training process. Data augmentation is also used to address data imbalance issues as in the case of BreakHis. An illustrative example of different data augmenters applied to a BreakHis image is presented in the figure 2.3.



(a) The original slide

(b) The same slide rotated with 40°

(c) The same slide flipped

(d) The same slide randomly distorted

(e) The same slide zoomed in ×10

(f) The same slide zoomed out ×10

Figure 2.3: An original ×100 slide extracted from a malignant (MC) patient with ID:16456 and five generated images from this slide using different data augmentation methods

2.4 BreakHis Computer-aided diagnosis systems overview

Over the past three years, several BreakHis based CAD systems have been built. To survey all this research we proposed a new taxonomy that organizes BreakHis related works into four different groups according to their adopted reformulation of this classification problem. The main motivation behind the proposed taxonomy, is to understand all the proposals that addressed BreakHis dataset for building CAD systems, their strengths and their weaknesses. The adopted taxonomy, its four groups and the works that belong to each group are summarized in Table 2.3.

Related works:	Reformulated as	Number of works
[169] [168] [24] [167] [166] [87] [36] [193] [164] [62] [2] [149] [64] [128] [154] [25] [210] [174] [23] [209] [26] [188] [122] [171] [204] [123] [150] [17] [89] [63] [141] [44] [39] [4] [7] [124] [35] [94] [129] [121]	MSB	40
[14] [61]	MIB	2
[50] [9] [67] [127]	MSM	4
Explored for the first time in this thesis	MIM	-

Table 2.3: BreakHis related works and their corresponding reformulations in our taxonomy.

Where MSB reformulation:classifies the input image as benign or malignant depending on its magnification factor. MIB reformulation:classifies the input image as benign or malignant regardless of its magnification factor. MSM reformulation:classifies the input image into one of the eight subcategories with taking into consideration its magnification factor. MIM reformulation:classifies each image into one of the eight subcategories regardless of its magnification factor. To illustrate the proposed taxonomy, we present in figure 2.4 the inputs, classifiers and outcomes of each reformulation:

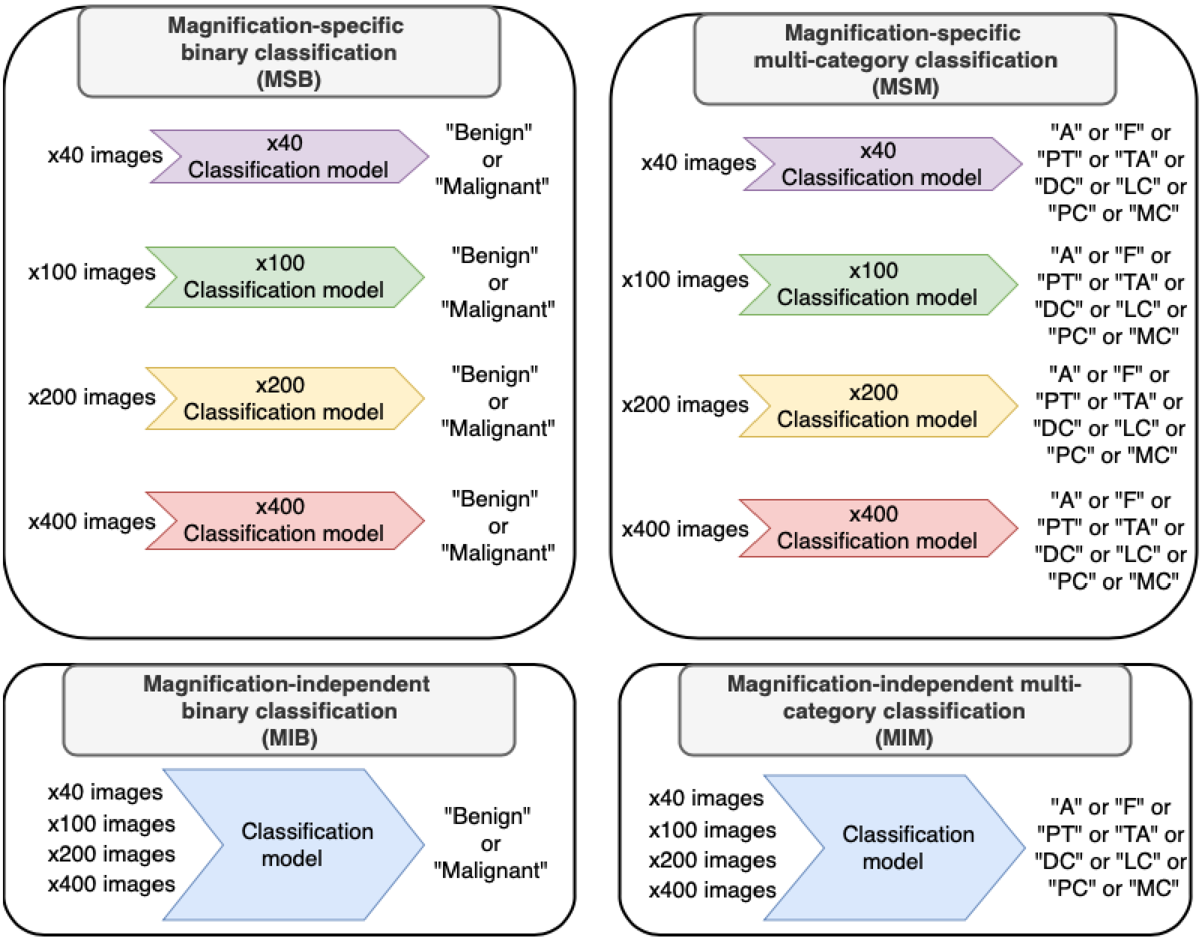

Magnification-specific binary classification (MSB)
x40 images
x40 Classification model
"Benign" or "Malignant"
x100 images
x100 Classification model
"Benign" or "Malignant"
x200 images
x200 Classification model
"Benign" or "Malignant"
x400 images
x400 Classification model
"Benign" or "Malignant"
Magnification-specific multi-category classification (MSM)
x40 images
x40 Classification model
"A" or "F" or "PT" or "TA" or "DC" or "LC" or "PC" or "MC"
x100 images
x100 Classification model
"A" or "F" or "PT" or "TA" or "DC" or "LC" or "PC" or "MC"
x200 images
x200 Classification model
"A" or "F" or "PT" or "TA" or "DC" or "LC" or "PC" or "MC"
x400 images
x400 Classification model
"A" or "F" or "PT" or "TA" or "DC" or "LC" or "PC" or "MC"
Magnification-independent binary classification (MIB)
x40 images
x100 images
x200 images
x400 images
Classification model
"Benign" or "Malignant"
Magnification-independent multi-category classification (MIM)
x40 images
x100 images
x200 images
x400 images
Classification model
"A" or "F" or "PT" or "TA" or "DC" or "LC" or "PC" or "MC"

Figure 2.4: An illustration of each reformulation in the proposed taxonomy

MSB, MIB and MSM works will be presented in sections 5.4, 5.5 and 2.7 respectively, while MIM will be explored for the first time in this paper in section 3.3. A work that belongs to two different groups will be reported only within the group that represents its main reformulation, while the results of its both reformulations will be included separately in their corresponding tables. Namely, MSM is the main reformulation of [9, 50, 67] while MSB is their secondary reformulation. Besides, the main reformulation of [24] is MSB while its secondary reformulation is MSM.

2.5 Magnification-specific binary classification works (MSB)

In this section, we will report all works that used an MSB approach. In fact, 85% of BreakHis CAD systems adopted this reformulation. These MSB works will be organized as follows:

- Section 2.5.1 summarizes works that adopted a traditional handcrafted based model
- Section 2.5.2 presents works that used deep learning based models
- Section 2.5.3 covers works that brought their contributions to the preprocessing phase
- Section 2.5.4 describes works that developed their CAD systems with a content-based histopathological image retrieval approach

- Section 2.5.5 highlights works that focused on domain adaptation

Then, in Table 2.4 we will summarize all these works and for each one we will present its best results, the adopted pre- and post-processing approaches, the used model and the learning strategies. Depending on the experimental setup used in each paper, results are going to be presented either as a mean value with a standard deviation over various trials or as a unique trial value. Depending on their availability, results are reported at different metrics levels including PLA, ILA, AUC and F1-score. For the seek of space, we used an abbreviation for each method, and the corresponding dictionary can be found in **??**.

2.5.1 Handcrafted descriptors based models

First BreakHis CAD systems adopted a traditional dual-stage approach, by extracting handcrafted features from the images, then using them to train a standalone classifier. At features extraction phase, some works evaluated multiple descriptors with the aim to select the most representative ones, while others used an unique descriptor. In this part, we will present all these works with their adopted image descriptors:

First, BreakHis authors explored in [169] the effectiveness of six state-of-the-art handcrafted features descriptors: Local Binary Patterns (LBP) [132], its variant Completed Local Binary Pattern (CLBP) [60], Local Phase Quantization (LPQ) [133], Gray Level Co-Occurrence Matrices (GLCM) [69], Parameter-Free Threshold Adjacency Statistics (PFTAS) [32] and Oriented FAST and Rotated BRIEF (ORB) [147], associated with four different classifiers: 1-Nearest Neighbor (1-NN) [194], Quadratic Linear Analysis (QDA) [181], Support Vector Machines (SVM) [21], and Random Forests of decision trees [101]. Then, To evaluate the effectiveness of fractal dimension [110] as the only descriptor; authors in [24] trained an SVM classifier with the fractal dimension of each image. Results of which demonstrated that using fractal dimension as a unique descriptor is more suitable when classifying ×40 images with a lot of self-similarities, but meaningless in higher magnification images with less self-similarities. As an additional part of this work, a multi-category classification task was elaborated with 16 experiments, each one classifying a benign and a malignant sub-classes.
Afterwards, authors in [149] evaluated different handcrafted descriptors in conjunction with a k-NN classifier, including: LBP, GLCM, PWT) and TWT. After that, authors in [26] proposed to use Zernaike moment, image entropy and fractal dimension features through a signal processing multilevel iterative Variational Mode Decomposition (VMD) [43], and select the most relevant features using Relief [92], before classifying them with a Least squares support vector machine (LS SVM). In [87], authors tried to enable an L1-norm Sparse SVM (SSVM) [212] to select the most relevant features from BreakHis images. According to their work, L1-norm is inconsistent in establishing features selection precisely and the SSVM could be biased towards large hyper-plane coefficients. To enhance its features selection quality, they assigned a weight to each feature depending on its Wilcoxon rank sum [102]. Lately, authors in [150] evaluated KAZE features [3] performance in a bag-of-features approach. Then, transformed these features into histogram information using an approximate Nearest Neighbour algorithm, and used them to train an SVM classifier.

Limitations Results achieved by different traditional handcrafted features were considered relatively acceptable as preliminary results but highly unstable. In fact, the major drawback of these traditional approaches is that: the model's quality depends on the extracted features, while acquiring a highly representative features is a very complex task. One of the main difficulties is the right descriptor choice, and even when various descriptors are combined together to increase their discriminative power, or post-transformed to select the most appropriate ones, their achieved results remains relatively low and unstable between different magnification levels.

2.5.2 Deep learning based classification models

To mitigate traditional practices limitations, some authors thought of entrusting features extraction as well as classification tasks to deep learning models giving their ability to select directly the most significant global features. These researches tried to successively boost their deep learning based approach, starting with the exploration of various models, to the analysis of different learning and adaptation strategies. In this part, we will organize these works according to their contributions at different deep learning aspects:

From traditional towards deep learning approach BreakHis authors were the first researchers to move towards the evaluation of a deep learning based CAD system for this dataset, by entrusting features extraction and classification tasks to a CNN model in [168]. They started by evaluating LeNet [97], but its results were lower than those reported by their previous traditional based model in [169]. Therefore, they opted for AlexNet [93] as a relatively deeper network.

Deep learning models trained with handcrafted features Generally, CNN models are provided with raw images as input. However, authors [122] were convinced by the importance of textural and pixel distributions contained in handcrafted features such as LBP or histogram descriptors. By consequence, they evaluated CNNs provided with various handcrafted features in comparison to those provided with raw images. After exploring different combinations, their best results were achieved with a model called (Model1-CNN-CH). The latter was a CNN model with residual blocks inspired from ResNet [70], and provided with a concatenation of extracted local-features using Contourlet Transform (CT) [41] and Histogram information descriptors.

CNN models comparison To find the adequate CNN for this classification task, authors in [174] compared the performance of three different CNN models: CaffeNet which is an AlexNet variant, GoogleNet [176] and ResNet-50. Results of which proved the efficiency of ResNet, the necessity of data augmentation, fine-tuning all layers, providing this CNN with large WSIs instead of small patches, and using ensemble learning by combining different magnification-specific models. In our last work [17], we explored the performance of another CNN which is a GoogleNet variant called Inception-v3 [177], and our results shown the efficiency of this CNN in comparison to shallower ones used in previous works.

Pre-trained CNN for features extraction use The improvement brought by the first CNN evaluations in BreakHis [168], encouraged its authors to explore further deep learning capabilities. They evaluated in [167] a transfer learning strategy with a pre-trained AlexNet and DeCAF features extraction approach [42]. The latter consists of extracting features from the pre-trained AlexNet's last layers, then using them to train a standalone classifier. Afterwards, authors in [23] explored the impact of three different dimensionality reduction methods on features extracted from a pre-trained VGG [161]: Principal Component Analysis(PCA) [145], Gaussian Random Projection(GPR) [19] and Correlation-Based Feature Selection(CBFS) [116].
Generally, when a pre-trained CNN is adopted as a features extractor, only its final layers features are exploited. To evaluate the potential contained in every layer of a pre-trained DenseNet-169 [82], in conjunction with XGBoost classifier [28], authors in [63] proposed a sequential features extraction framework. This evaluation proved that the last convolutional layers provide more significant features than the final fully connected layers. Another interesting finding of this work is that: lower level layers contribute significantly to ×40 images classification. Similarly, mid range magnifications ×100 and ×200 are better represented by mid level features. While ×400 images are better captured by higher level layers.

Pre-trained CNN for Fine-tuning use For fine-tuning, various practices are adopted. In some works, all the pre-trained CNN layers are fine-tuned; while in others, only the last fully connected layers are retrained. Authors in [210] proposed a dual-stage fine-tuning method, which retrained only the fully connected layers in a first place, then the whole network. To justify this choice they also evaluated it against each one of its two independent stages.

Features extraction versus fine-tuning use In [39], the authors demonstrated that fine-tuning the three last layers of a pre-trained AlexNet is more efficient than SVM classification of concatenated features extracted from two pre-trained CNNs (AlexNet and VGG16).

Features extraction combined with fine-tuning An ImageNet pre-trained CNN extracted features are meant to be the common high level features between ImageNet and BreakHis classification tasks. However, the used CNN is not supervised to extract necessary features for BreakHis classification. Thus, a gap is generated between the extracted features and the required specific domain features [100]. To mitigate this gap, [204] proposed a hybrid transfer learning approach called deep domain knowledge-based features model, by adding a preliminary knowledge adaptation step that consists of retraining (fine-tuning) the pre-trained CNN on BreakHis classification task in a first place for more efficient features extraction.

CNN Features extraction versus handcrafted features extraction Authors in [7] evaluated features obtained with traditional handcrafted descriptors in comparison with those extracted from a pre-trained AlexNet. Surprisingly, LBP handcrafted features have proven to be slightly better than AlexNet features. However, this comparison remains very restrictive with a relatively shallow CNN which was found also in [168] to be barely capable of outperforming handcrafted based models.

CNN features post-encoding using Fisher Vector Post-encoded CNN features using Fisher Vector (FV) [138] are known for their good classification potential [30, 52]. To evaluate their performance in BreakHis problem, authors in [166] started with fine-tuning an pre-trained VGG model as a preliminary adaptation of this CNN on BreakHis classification. Then extracted a dense set of local features from its last convolutional layer in order to encode them into FV descriptor. Afterwards, the same authors presented a second work [164] with the aim to overcome FV high dimensionality issue. In fact, they proposed a supervised intra-embedding model designed to embed each block of the FV descriptor into a lower dimensional feature space, using a dimensionality reduction algorithm based on a multilayer neural network. In their next work [163], they proposed new features representation method called Component Selective Encoding (CSE). Therefore, they used the same pre-trained VGG as in [166], along with a new adapted dimensionality reduction method inspired from one of their previous works [165]. The latter aims to reduce each FV component individually, unlike [164] where they were reduced uniformly. This adaptation was justified by the fact that some regions in these images are more relevant than others.

CNN architecture adaptation Inspired by the effectiveness of GoogleNet inception module in capturing multi-scale features with different convolutions, Authors in [2], proposed a new version of this module called ”Transition module” and integrated it to AlexNet. This new version was designed with the aim to ease the abrupt transition between the last convolution layer and the first fully connected layer. Unlike inception module, no prior dimensionality reduction was included to this transition module. Then, authors in [128] proposed a new CNN architecture composed of fifty convolutions, and compared it to several handcrafted features based models. Afterwards, authors in [193] designed a new CNN called BiCNN inspired from GoogleNet architecture. To use this CNN in the binary classification task they proposed to take into consideration sub-class information in conjunction with binary labels of each image as a prior knowledge. They claimed that this consideration of both annotation levels could help the proposed model to better learn features distance between binary classes. In [188], the authors tried to leverage recent findings in rotation equivariant CNNs [33] with the inherent symmetry under rotation and reflection of histopathological images, in order to build a rotation and reflection equivariant CNN inspired from DenseNet. Lately, another work [94] proposed a CNN model composed of different layers combinations (convolutional, pooling and fully connected layers), whereas various compositions and hyper-parameters were evaluated to determine the most adequate architecture for BreakHis classification.

Multiple Instance Learning with CNN Practically, CNNs requires original WSIs images to be resized to fit in their input layer. Some practitioners prefer to extract low size patches from original WSIs in order to avoid loosing any discriminative information. However, adopting a patch based approach is very challenging, since only a small number of extracted patches is correctly labeled. This fact is due to the presence of benign areas in malignant WSIs. To address this mislabeled patches issue, authors in [171] proposed to use a Multiple Instance Learning (MIL) approach with randomly extracted 64×64 patches. In fact, they noticed that BreakHis distribution at patient and image level is similar to MIL reasoning, and adapted its formulation to two different settings. The first one meets the labeling at image level, where each image was considered as a bag of instances. The second setting considered each patient as a bag. They explored twelve different MIL methods including recent ones such as deep learning based MIL-CNN [175] and non-parametric MIL [189]. In another MIL based approach [35], the authors introduced a new MIL CNN layer termed Multiple Instance Pooling (MIP) layer with the aim to select from each bag their most discriminative instances with the higher feature responses, instead of capturing all their instances. This constraint was integrated into the loss function by considering only the loss associated to higher activation instances.

Deep active learning To avoid mislabeled patches when adopting a patch based approach, practitioners are forced to annotate all extracted patches. However, this task is expensive, very tedious and time-consuming. In order to reduce this labeling burden, authors in [44] proposed a deep active learning framework enhanced with a boosted confidence approach. This approach is based on an active learning model which is firstly initialised with very limited labeled data. Then, it selects at each iteration the lowest confidence unlabeled samples (highest entropy samples) and give them to domain experts for annotation. By consequence, it reduces considerably the annotation cost. However, it ignores the less representative samples (higher confidence samples with lower entropy) and their potential. Therefore, authors in [44] provided these remaining samples to the model itself for auto-annotation without any additional manual-annotation cost.

Ensemble learning To prove the effectiveness of ensemble learning with features learned by different classifiers at various scales, authors in [36] explored the performance of an ensemble of different magnification-specific model where each one is a pre-trained GoogleNet. In fact, these CNNs were trained separately in a magnification-specific way, but for each test image an ensemble of all these magnification-specific CNNs was aggregated using a majority voting rule.

Deep Belief Network Inspired by the outstanding results achieved with Deep Belief Networks (DBN) in many fields applications [202], researchers in [124] used a Deep Belief Network (DBN) composed of four stacked Restricted Boltzmann Machine (RBM). But, instead of using raw images they provided it with handcrafted Tamura features [178].

Autoencoder In [141], the authors proposed a hybrid framework that starts with a LandMark ISOMAP (L-ISOMAP) embedding [180] to extract the most significant features in BreakHis images, followed by an SSAE with two stacked sparse Autoencoders and a classification output layer. This choice was motivated by the fact that Autoencoders have shown distinguishable results in different image classification tasks [203]. In their results, they stated that this approach allowed them to achieve an improvement in classification rate while reducing the overall computational cost.

Limitations Results achieved by different deep learning models are considerably higher than those presented with traditional approaches. Nevertheless, deep learning models are extremely data-hungry and require a large amount of data, while medical applications such as breast cancer diagnosis always suffer from a lack of data. To mitigate this limitation, often practitioners are forced to adopt artificial data augmenters as a preprocessing. In addition, determining the best hyper-parameters for this kind of models is a black art with no guiding theory. Moreover, unlike handcrafted engineered models where what is learned is easy to comprehend, deep learning approaches are not able to give users feedback or interpretability on the discriminative features used to decide about each patient diagnosis. Besides breast cancer, many deep learning based medical applications exists and each one of them has different limitations [151]. For a full review on this topic, we refer the reader to the following recent references [8, 20, 27, 49, 75, 109, 113, 120, 155]

2.5.3 Preprocessing methods

To build these CAD systems either with deep learning or traditional approaches, raw images need various preprocessing transformations. In this part we will present works where the main contribution is related to the preprocessing phase:

Data augmentation Deep learning models, and especially CNNs requires an important volume of training data and particularly when they are to be fine-tuned. Thus, to generate a sufficient number of data samples for fine-tuning a pre-trained inception v3 on ×40 images classification, authors in [25] evaluated different data augmentation techniques and reported their results.

Clustering as a preprocessing To explore the hidden similarities in morphological textures of BreakHis images, authors in [123] adopted a clustering algorithm as a preprocessing step. This method aims to extract statistical and geometrical clusters hidden in the data structure. In order to prove the efficiency of their approach, they evaluated the performance of a CNN provided with these cluster-transformed images using various clustering algorithms in comparison to a CNN provided with raw images. In [154], authors evaluated a segmentation preprocessing step based on a clustering algorithm to highlight nuclei regions in each image before extracting their features and provide them to different classifiers. Lately, in [89], the authors used a K-means clustering on each image to highlight its nuclei segments, before extracting entropy features from these cluster-transformed images using Discrete Wavelet Transform (DWT) [157]. Then, evaluated an SVM classifier trained with these features.

Stain normalization To address the stain variability of BreakHis images, authors in [62] were motivated by learning the color-texture variation of these images instead of reducing the color-variations between them. Therefore, they explored several combinations of various color-texture descriptors along with different classifiers. After identifying the best performing features-classifier combination in each magnifications subset, they combined them in an integrated model. Then, the same authors tried in [64] to provide indications about whether it is possible for a model to learn this color-texture variability instead of normalizing it. Their experiments found that: on one hand; stain normalization could be substituted by joint color-texture features learning to achieve higher results, on the other hand: gray scale transformation is not a good stain normalization method and could decrease the classification accuracy. Recently, authors in [121] started from the conviction that conventional normalization techniques amplify the existing noise in images when applied directly, and propose a normalization strategy that includes a noise amplification control step.

2.5.4 Content-based histopathological image retrieval CAD systems

Unlike standard CAD systems, in a Content-based Histopathological Image Retrieval (CBHIR) system, the model search in the dataset for other images with similar content to the query one, and return them to the pathologist to use their diagnostic information as reference. In this part we will present works that adopted a CBHIR system:

One of the first attempts to employ a CBHIR approach is [209]. Authors in this work were aware of the fact that a CBHIR system is computationally intensive especially when retrieving from a large-scale dataset, because it is essentially based on feature vectors comparison to measure images similarities. To address this aspect, they proposed a three-steps framework: Firstly, all images in the dataset undergone a binarization encoding with different tile sizes in order to decrease the required computations and ensure their size-scalabality. Then, for each query image a similarity-based search is conducted to look for the closest proposals in the encoded dataset. Finally, the retrieved images are ranked, giving their similarities to the query image using Hamming distance. To reduce retrieving time, authors in [129] used a pre-trained VGG-19 that extracts class-specific and patient-specific tumorous descriptor simultaneously. In fact, they divided the adopted framework into two different phases. In the first phase, the pre-trained VGG-19 extracts the last layer features as a 1000-dimensional vector from each image in the gallery set (the training set) and then uses these features to train a multi-patient classifier which gives for each image a score evaluating if it belongs to a given patient (82-dimensional vector) and a binary malignancy classifier to determine either is benign or malignant (2-dimensional vector). In the second phase, they extracted from each query image its 1000-dimensional features using a pre-trained VGG-19, then these features are passed through the fine-tuned multi-patient and binary classifiers to find a conjoint patient/class 2-dimensional vector. This vector is used to retrieve similar images from the gallery set.

2.5.5 Domain adaptation approach

Mostly, all BreakHis models are built with the assumption that the distribution of training and testing data are the same, whereas; others claimed that this assumption is not correct since histopathological images are prepared and stained in different laboratories with different standards, which could adversely degrade the classification rate. In this part we will present works that proposed a domain adaptation approach:

Authors in [4] were the first and only to address the aforementioned issue; they proposed a new learning framework with an unsupervised domain adaptation approach based on the data representation-learning. The goal of this domain adaptation is to reduce the differences between the marginal distributions of source and target domains while learning a new representation for both domains. This method is based on the creation of an invariant space where training and test sets are projected to adapt their different domains.

Work	Preprocessing	Patch/Slide	Features extractor	Classifier	Transfer learning	Training/Test	Metrics	Results(%)			
								×40	×100	×200	×400
[169]	None	WSI	PFTAS	QDA	None	70 % / 30 %	PLA	83.8 ± 4.1	82.1 ± 4.9	84.2 ± 4.1	82.0 ± 5.9
[168]	-Res(350x230) -SMI	-Rnd(32x32, 64x64) -SW(32x32, 64x64)	Emax(AlexNet)		ImageNet	70 % / 30 %	ILA	85.6 ± 4.8	83.5 ± 4.8	84.6 ± 4.2	86.1 ± 6.2
							PLA	90.0 ± 6.7	88.4 ± 4.8	84.6 ± 4.2	86.1 ± 6.2
[24]	Binarization	WSI	FD	SVM	None	50 % / 50 %	F1	97.9	16.5	16.5	25.3
[167]	None	MKV(1,4,6)	CaffeNet	LR	ImageNet	70% / 30%	ILA	84.6 ± 2.9	84.8 ± 4.2	84.2 ± 1.7	81.6 ± 3.7
							PLA	84.0 ± 6.9	83.9 ± 5.9	86.3 ± 3.5	82.1 ± 2.4
[67]	-DAB(IV, Rot, Tr, Flip)	WSI	NDCNN(GoogleNet)		ImageNet	50% / 50%	ILA	95.8 ± 3.1	96.9 ± 1.9	96.7 ± 2.0	94.9 ± 2.8
							PLA	97.1 ± 1.5	95.7 ± 2.8	96.5 ± 2.1	95.7 ± 2.2
[166]	None	WSI	FV(VGG)	SVM	ImageNet	70% / 30%	ILA	87.0 ± 2.6	86.2 ± 3.7	85.2 ± 2.1	82.9 ± 3.7
							PLA	90.0 ± 3.2	88.9 ± 5.0	86.9 ± 5.2	86.3 ± 7.0
[87]	None	WSI	DR(ASSVM, WRS)		None	70% / 30%	PLA	94.97	93.62	94.54	94.42
[36]	-Res(370x230) -DAB(Rot, Flip)	Rnd(224x224)	GoogleNet		ImageNet	80% / 20%	ILA	94.82	94.38	94.67	93.49
[193]	-SMI -DA(Rot, Scal, Mir)	WSI	NDCNN(GoogleNet)		ImageNet	75% / 25%	ILA	97.89	97.64	97.56	97.97
							PLA	97.02	97.23	97.89	97.50
[164]	None	WSI	DR(FV(ConvNet), MNN)	SVM	ImageNet	70% / 30%	ILA	87.7 ± 2.4	87.6 ± 3.9	86.5 ± 2.4	83.9 ± 3.6
							PLA	90.2 ± 3.2	91.2 ± 4.4	87.8 ± 5.3	87.4 ± 7.2
[62]	None	WSI	Integrated		None	70% / 30%	ILA	88.09			
							PLA	88.40			
[2]	None	P(228x228)	NDCNN(AlexNet,trans)		None	Not specified	ILA	82.7%	Not evaluated	Not evaluated	Not evaluated
[149]	None	None	PWT	KNN	None	75% / 25%	ILA	Not evaluated	Not evaluated	Not evaluated	85.62
[64]	RGBT	WSI	JCTF	Linear SVM	None	70% / 30%	PLA	86.88 ± 2.37	88.41 ± 2.73	88.86 ± 3.76	87.55 ± 3.01
[128]	-Res(350x230) -SMI -DA(Rot, Flip)	WSI	NDCNN		None	70% / 30%	ILA	77.5	Not evaluated	Not evaluated	Not evaluated
[154]	-ETB -NDS	WSI	PFTAS	RF	None	Not specified	ILA	81.7 ± 2.8	81.2 ± 2.7	80.7 ± 3.4	81.5 ± 3.1
[25]	- DA(Rot, Mir, Dis)	WSI	Inception v3		ImageNet	70% / 30%	ILA	0.86	Not evaluated	Not evaluated	Not evaluated
[210]	-DA(Zoom, Flip)	Rnd(224x224)	Emax(VGGNet)		ImageNet	80% / 20%	ILA	91.28	91.45	88.57	84.58
[174]	-Res(350x230)	WSI	Eavg(ResNet)		ImageNet	Not specified	PLA	95.0 ± 3.64			
[23]	-Res(224Ã 224) -GSC	WSI	VGG	NN	ImageNet	75% / 25%	ILA	84.0	88.2	87.0	80.3
[209]	None	SQ	BE	SBC	None	70% / 30%	ILA	47.0	40.0	40.0	37.0
[26]	MVD	WSI	DR((Zer, FD, Ent),Rlf)	LSSVM	None	70% / 30%	PLA	87.7	85.8	88.0	84.6
[188]	None	Rnd(64x64)	NDCNN(DenseNet)		Camelyon	75% / 25%	ILA	96.1 ± 3.2	Not considered	Not considered	Not considered
[122]	None	WSI	CT, HI+KM	NDCNN	None	Not specified	ILA	94.40	95.93	97.19	96.00
							F1	95.00	97.00	98.00	96.00
[171]	None	Rnd(64x64)	PFTAS	NPMIL	None	70% / 30%	ILA	87.8 ± 5.6	85.6 ± 4.3	80.8 ± 2.8	82.9 ± 4.1
							PLA	92.1 ± 5.9	89.1 ± 5.2	87.2 ± 4.3	82.7 ± 3.0
[204]	-Res(224x224) - DA(CROP, ANP, FLIP)	WSI	ResNet-152, GoogleNet	SVM	ImageNet	80% / 20%	ILA	81.2 ± 2.5			
[123]	KM	WSI	NDCNN		None	Not specified	ILA	85	90	90	90
							F1	93	93	92	93
[150]	GSC	WSI	HI, KAZE	SVM	None	70% / 30%	ILA	85.9 ± 1.6	80.4 ± 1.4	78.1 ± 2.2	71.1 ± 3.3
							PLA	86.4 ± 2.2	81.6 ± 1.6	77.8 ± 1.6	72.9 ± 2.8
							F1	90.2 ± 1.1	86.5 ± 1.0	84.6 ± 2.2	79.9 ± 3.2
							AUC	94	Not specified	Not spcified	Not specified
[17]	SMI	WSI	Inception-V3		ImageNet	70% / 30%	ILA	86.5 ± 3, 7	83.2 ± 2.4	85.4 ± 0.7	80.3 ± 2.2
							PLA	87.6 ± 3.9	82.4 ± 2.7	86.1 ± 0.7	79.7 ± 3.2
							F1	93.0	88.9	89.4	86.9
[89]	KM	WSI	DWT	LSVM	None	Not specified	ILA	93.3			
[63]	-Res(224x224) -DA(Rot, Flip, HS,WS,TR)	WSI	DR(DenseNet-169,XGB)	XGB	ImageNet	70% / 30%	PLA	94.71 ± 0.88	95.9 ± 4.2	96.76 ± 1.09	89.11 ± 0.12
[141]	-GSC -DA	WSI	DR(WSI,L-Isomap)	SSAE	None	Not specified	ILA	96.8	98.1	98.2	97.5
[50]	-Res(341x224) - SN - DA(Rot, Flip, WS, HS) -UP	SW(224x224)	ResNet-152		ImageNet	70% / 30%	ILA	98.6	97.9	98.3	97.6
			Emdt(ResNet-152)		ImageNet	70% / 30%	PLA	98.77			
[44]	None	WSI	AlexNet		ImageNet	70% / 30%	ILA	90.69	90.46	90.64	90.96
[39]	Res(227x227)	WSI	AlexNet		ImageNet	Not specified	ILA	90.96 ± 1.59	90.58 ± 1.96	91.37 ± 1.72	91.30 ± 0.74
[4]	None	WSI	DR(PFTAS,PROJ)	QDA	None	70% / 30%	PLA	89.1 ± 2.6	87.3 ± 3.8	88.4 ± 3.6	86.6 ± 2.8
[7]	None	WSI	LPQ	SVM	None	70% / 30%	ILA	91.1	90.7	86.2	84.3
							AUC	0.96	0.96	0.93	0.90
[124]	CE	WSI	Tam	DBN	None	70% / 30%	ILA	88.7	85.3	88.6	88.4
[35]	-Res(370x230) -DA	P(224x224)	NDCNN		None	80% / 20%	ILA	89.52	89.06	88.84	87.67
[94]	None	P(64x64, 32x32)	NDCNN		None	70% / 30%	ILA	82 ± 2.8	86.2 ± 4.6	84.6 ± 3	84 ± 4
							PLA	83 ± 3.2	81 ± 4.2	84.2 ± 3.4	81 ± 2.4
[129]	-Res(224x224) -SMI	WSI	VGG-19	E(SVM)	ImageNet	98% / 2%	ILA	80			
[9]	DA(Rot, Flip)	WSI	Eiter(NDCNN)		None	70% / 30%	ILA	98.33	97.12	97.85	96.15
[121]	SN	WSI	HI	NDCNN	None	85% / 15%	ILA	95.0	96.6	93.50	94.2

Table 2.4: MSB classification models and results.

2.6 Magnification-independent binary classification works (MIB)

After reporting MSB classification works and their results, we will present in this section binary classification models that adopted a magnification-independent approach (MIB). In fact, only two works belongs to this group, and we will present them as follows:

- Section 2.6.1 presents the first attempt that introduced magnification-independent training to binary classification

- Section 2.6.2 describes the work that explored the discriminative value contained in each magnification factor subset.

Afterwards, we will summarize in table 2.5 the best model of each work and its achieved results.

2.6.1 From a magnification-specific towards a magnification-independent approach

Magnification-independent approach for BreakHis based models was first introduced in [14], where authors proposed to classify histopathological images as either benign or malignant regardless of their magnification factors. To evaluate their MIB reformulation, they elaborated two experiments. In the first one they explored a single task CNN, while in the second one they trained a multi-task CNN. The single task CNN was trained on all magnification subsets combined and tested on each magnification independently to allow its direct comparison with previous magnification-specific works. Results of this comparison proved that this magnification-independent model outperformed some previous magnification-specific works, and more importantly achieved stable results over different magnification. In their second experiment, the authors explored the performance of a multi-task version of the first adopted magnification-independent CNN. This multi-task version was equipped with an additional classifier which serves to predict the magnification level of each input image. Its binary classification results decreased slightly in comparison to those obtained with the single task version. This drop in accuracy was justified by the usefulness of the added magnification factor classifier for the binary classification task.

2.6.2 Cross-magnification evaluation

Authors in [61] tried to evaluate the discriminative value of each magnification subset independently with a cross-magnification training/test schema. In other words, they tried to explore all possible training/test combinations, where each time a model is trained on a given subset and tested on this same subset or another one. Each model adopted a dual-stage framework. Firstly, it extracted color-texture features. Then, it provided a concatenation of these features to train a majority voting ensemble composed of: SVM , Nearest neighbors, Decision tree and Discriminant Analysis. After evaluating all possible cross-magnification data splittings, these experiments revealed some interesting findings: Models trained with extreme magnification subsets (×40, ×400) achieved lower classification results with a large variation between different magnification test sets, while those trained with mid-range magnification subsets (×100, ×200) obtained higher results and proved to be more stable over all magnification test sets. The authors justified these facts by the large variability in morphological textures of extreme magnification images (×40, ×400) in comparison to those captured with mid-range magnifications (×100, ×200).

Work	Preprocessing	Patch/Slide	Features extractor	Classifier	Transfer learning	Training/Test	Metrics	Results(%)			
								×40	×100	×200	×400
[14]	-Res(460x460) -DA(Rot, Crop, Flip)	Rnd(100×100)	NDCNN		None	70% / 30%	PLA	83.08 ± 2.08	83.17 ± 3.51	84.63 ± 2.72	82.10 ± 4.42
[61]	None	WSI	JCTF	Emv(classifiers)	None	70% / 30%	PLA	87.2 ± 3.74	88.22 ± 3.28	88.89 ± 2.51	85.82 ± 3.81

Table 2.5: MIB classification models and results.

2.7 Magnification-specific multi-category classification works(MSM)

After reporting all binary classification reformulations, we will present in this section the multi-category classification models trained with a magnification-specific approach (MSM). Mostly, all MSM works were built within a dual-stage framework, where each stage is devoted either to multi-category or binary classification. Thus, these works will be reported according to the logical ordering of their two stages as follows:

- Section 2.7.1 presents works where the multi-category classification is elaborated before the binary classification
- Section 2.7.2 describes works where the binary classification is elaborated before the multi-category classification
- Section 2.7.3 reports works where both classification stages are elaborated independently
- Section 2.7.4 outlines works where the multi-category classification is elaborated without any binary classification stage

Then, in table 2.6 we will report the best achieved results in each work.

2.7.1 Multi-category classification stage before binary classification stage

The first CNN that was designed essentially for multi-category classification is [67]. The latter proposed a two-stages framework composed of a multi-category classification model followed by a binary classification phase. The used CNN was trained only with a multi-category classification output. Then, the binary classification output was intuitively deduced from its corresponding sub-class. Authors of this work noticed that this reformulation has never been explored before due to the high similarities between different sub-classes. According to them, the variance between instances from the same sub-class are greater than the one between those from different sub-classes. To overcome this issue, they integrated a learning constraint to the multi-category CNN. This constraint was added to the output loss function with the aim to control different features similarities during the training, by minimizing the euclidean distance between instances from the same sub-class, while maximizing it between those from different sub-classes or main classes. In addition to the proposed constraint, the authors in this work explored different configurations and also proved the necessity of data augmentation and transfer learning approaches when training a deep learning CNN.

2.7.2 Binary classification stage before multi-category classification stage

Authors in [50] firstly fine-tuned an ImageNet pre-trained ResNet-152 for the binary classification task. Then, retrained the same fine-tuned ResNet-152 with a multi-category classification output layer. The second stage was composed of two modules, the first one was devoted for benign sub-classes and the second one for malignant sub-classes. After identifying its main class in the binary classification stage, each instance was provided to its corresponding module for sub-class identification. An image was considered correctly classified if only its both stages outputs were correctly classified. For both stages, the decision was made at two levels: At image level; the decision was obtained by merging all extracted patches outputs using majority voting rule. At patient level; the decision was elaborated with a Meta-decision Tree (MDT) [182] which acts like a trainable ensemble learning approach combining all magnification-specific models. Furthermore, this work revealed that the improvement brought by stain normalization and data augmentation is around 13%.

2.7.3 Multi-category classification and Binary classification tasks performed independently

In [9], the authors tried to elaborate each one of these two classification tasks independently without any logical link between them. For each classification task, they compared a new-designed CNN to a handcrafted features based model. In the handcrafted features based approach, they evaluated various descriptors encoded with two coding models (bag of words and locality constrained linear coding) in conjunction with an SVM classifier. For the CNN based model, they evaluated a new designed CNN in different use cases: firstly, as a global end-to-end CNN, then as a features extractor or a final classifier provided with prior extracted handcrafted features. Results of this comparison proved the efficiency of the end-to-end CNN model in both classification tasks. Afterwards, they explored the improvement brought by data augmentation as well as ensemble learning with merging prediction outputs captured from the same model at ten different training iterations.

2.7.4 Multi-category classification without binary classification

Unlike previous works, [127] used a multi-category classification stage only. In fact, it evaluated the performance of three different CNNs in this classification task; namely, a ResNet-v1 model and two Inception variants v1 and v2, all trained in magnification-specific manner. Results of which revealed the ability of the pre-trained ResNet-v1 model to outperform the inception CNNs and achieve a recognition rate of around 95% in such a tedious task. This result was achieved in conjunction with different preprocessing methods including data augmentation and stain normalization, in addition to a fine-tuning strategy where all ResNet layers were retrained on BreakHis multi-category classification task.

Work	Preprocessing	Patch/Slide	Features extractor	Classifier	Transfer learning	Training/Test	Metrics	Results(%)			
								×40	×100	×200	×400
[24]	-Bin	WSI	FD	SVM	None	50 %/ 50 %	F1	96.4 over all magnifications			
[67]	-DAB(IV, Rot, Tr, Flip)	WSI	NDCNN		ImageNet	50% / 50%	ILA	92.8 ± 2.1	93.9 ± 1.9	93.7 ± 2.2	92.9 ± 1.8
							PLA	94.1 ± 2.1	93.2 ± 1.4	94.7 ± 3.6	93.5 ± 2.7
[50]	-Res(341x224) -SN -DA(Rot, Flip, WS, HS) -UP	SW(224x224)	ResNet-152		Im-Break	70% / 30%	ILA	95.6	94.8	95.6	94.6
			Emdt(ResNet-152)		Im-Break	70% / 30%	PLA	96.25 over all magnifications			
[9]	-DA(Rot, Flip)	WSI	Eiter(NDCNN)		None	70% / 30%	ILA	88.23	84.64	83.31	83.98
[127]	-SN(Macenko) -DA(Rot, Flip) -Res(224x224)	WSI	ResNet		ImageNet	80% / 20%	ILA	95.0 over all magnifications			

Table 2.6: MSM classification models and results.

2.8 Conclusion

In this chapter, we proposed a taxonomy that classifies BreakHis based CAD systems into four reformulations (MSB, MIB, MSM, MIM). Using this taxonomy, we provided a comprehensive survey of all CAD systems that used BreakHis dataset. We highlighted their main contributions especially when DL is adopted, their used preprocessing methods, their training strategies in addition to their achieved results at different evaluation levels and various metrics. We believe that all these summarized studies will serves to give an integral and complete idea of the existing literature on this this dataset to all upcoming researchers who wants to explore its potential. In the context of this thesis, the provided overview will serve us as a base analysis to the next chapter of this thesis to formulate and evaluate an ideal BreakHis based CAD system using deep learning. In fact, this analysis conducted us to identify and evaluate the most suitable reformulation for this problem from the clinical standpoint as well as the most adequate DL pre- and post-processing approaches. Using all these knowledge, we expose in the next chapter an ideal model for this problem which to our knowledge has never been explored or even identified in the literature before.

CHAPTER 3

An analysis of the ideal breast cancer computer-aided diagnosis system using BreakHis and deep learning

Contents

This chapter constitute the second part of the paper published in Neurocomputing [15] and the entire paper published in the Proceeding of LOPAL international conference [17].

3.1 Introduction and motivation

The current chapter is the continuation of the last chapter. Thus, the motivation behind the study elaborated in this chapter and the last one are similar. However, the novelty of the current chapter aims this time to build and analyze the ideal breast cancer CAD system for BreakHis using DL and and all learnt lessons from the overview elaborated in the previous chapter. As we presented in the last chapter, BreakHis based Breast cancer CAD systems were reported into four different groups giving their adopted reformulation for BreakHis images classification problem. Some works proposed to explore the binary classification aspect of this dataset to build a CAD system able to help pathologists decide whether a patient's slide is benign or malignant, while others preferred to leverage BreakHis potential in its multi-category annotation to find the exact malignancy sub-class for each image and give further information about the tumor severity and the required treatment. Each one of these two different classification tasks was either built with taking into consideration the microscopic magnification factor in the used images or regardless of this element. These different possibilities results in the four reformulation groups of the presented taxonomy in the last chapter. In the last chapter, we reported each paper in its corresponding reformulation group, we explored its main contribution, its used pre-processing methods, adopted model, post-processing and learning strategies and its achieved results at different evaluation levels. In this chapter, we will first, elaborate a comparison between binary and multi-category classification reformulations. Then, we will compare the magnification-specific to magnification-independent training approaches. This analysis helped us to identify the ideal formulation for this classification problem from a clinical as well as a practical standpoint. Then, we will build the corresponding model for this formulation using DL along with the best combination of pre- and post-processing methods that have shown to be the most efficient in the last chapter.

3.2 Comparison between different reformulations of BreakHis problem

3.2.1 Binary classification versus multi-category classification

In this part we will build our comparison between between binary classification and multi-category classification reformulations.
In fact, binary classification is deciding whether a given breast cancer lesion is benign or malignant, while multi-category classification is responsible for not only identifying whether a lesion is malignant but also determining the exact benign or cancer subtypes, as both benign and malignant breast lesions encompass different subcategories. Thus, it would be unfair to say that binary classification is independent from multi-category classification, and we prefer to say that it is inherently included in the latter. This inclusion formulation is always invoked in BreakHis related works that are devoted for multi-category classification. Most of these works are presented as two-stage framework [9, 24, 50, 67, 119] where results of each classification task are reported in each stage, except in [127] where the binary classification results are not explicitly reported but still can be implicitly concluded from their subcategories classification results.

This inclusion formulation leads us to intuitively consider the added value of the encompassing multi-category classification task as a natural comparison criteria. In fact, as depicted in [9,24,50,67,119,127], a multi-category classification has more clinical value than a simple binary classification because:

- First, finding the exact tumor sub-category provides more details about patients health conditions, which relieves the workloads of pathologists and guides them to make more optimal therapeutic schedules.

- Second, different treatment options are available for breast cancer patients and determining its subtype could be helpful in predicting the patient's response to a particular therapy; notably, invasive lobular tumor gains a clear benefit from systemic therapy when compared to invasive ductal [10].

- Third, the correct recognition of benign lesion type is also important because the patient's risk of developing subsequent breast cancer varies among different types of benign lesions [65].

In General, a framework for multi-category classification purpose with an included binary classification option could be performed either by one of the following manners:

- The first solution is applying the binary classification in the first stage before the multi-category classification phase as in [50]. In this case, a first model is dedicated to determine whether a test sample is benign or malignant, then depending on its main malignancy class this sample is given to the corresponding multi-category model in the second stage for further classification of its exact sub-class (the second stage contains a model for benign subcategories and a model of malignant subcategories).
- The second solution applying the multi-category classification in the first stage before the binary classification as in [67]. In this case, only one multi-category classification model is used for all the eight subcategories combined, then the binary main class is implicitly deduced from the unique multi-category model without the need of an additional model dedicated for the binary classification task.
- The third solution is applying the multi-category classification independently of the binary classification as in [9]. In this case, models from both tasks are applied independently, and neither the main class of a given sample is used to specify its subcategory or the opposite. In addition, one multi-category classification model could be used for all subcategories or two different multi-category classification models could be adopted.

For our proposed ideal model, we believe that the most intuitive and natural manner to build a multi-category classification framework with an included binary classification option would be a dual-stage approach as in the second solution: applying a multi-category classification in the first stage with one model for all eight subcategories and then deduce directly the main malignancy class of the tested sample (benign or malignant) in the second stage without any additional model to achieve this binary classification task. This choice is due to the fact that:

- The first solution requires three different models, resulting in more training and adaptation efforts. Moreover, in this first solution, any misclassification in the binary classification stage will directly impact the most important task which is the second stage (multi-category classification stage).
- The third solution requires two or three different models, resulting also in more training and adaptation efforts. In addition, this third is omitting and neglecting completely the logical linking between the two classification tasks as they are performed independently, and a test sample could be classified differently by the two classifications resulting in confusing verdicts.

3.2.2 Magnification-independent approach versus magnification-specific approach

To train such a multi-category classification model, one could take into consideration the magnification factor of each image or train this model regardless of the magnification feature. For our ideal model hypothesis, we believe that a training this model with magnification-independent approach is clinically and practically more suitable than adopting a magnification-specific approach, because to the following factors:

- First, in a magnification specific approach, one specific model is required for each magnification subset, resulting in four different models and subsequently more training and adaptation efforts.
- Second, during testing phase in a magnification-specific approach, magnification factor of each test image must be known and the exact corresponding model should be used. However, such specific information might not be available for all images.

- Third, In a magnification-specific approach, the classification model might perform poorly when test images are acquired at new magnification levels. Because, during training stage, classifiers learn magnification specific features and they could not adapt themselves to unseen image features.

- Fourth, unlike magnification-specific methods, a magnification-independent approach has the ability to directly benefit from additional training data, and such additional data could be captured with the same or different magnification factors than previous ones. In other words, this magnification-independent approach could benefit from additional labeled training data in a straightforward manner, notably for data augmentation.

- Fifth, when authors of magnification-specific models elaborated in [50, 174], employed at test phase ensembles with their four magnification-specific models, they all achieved noticeably improved results. Such a fact can be interpreted here as a reinforcement of our hypothesis claiming that training a model in magnification-independent manner with features from different scales could implicitly improve the model generalization capability.

- Sixth, a classification system which is intended to be used in a real clinical practice should handle the diversity in microscopic images and should not depend on any device settings such as magnification level, because this flexibility is necessary when deploying this system in underdeveloped, developing countries or in rural areas, which may have microscopes with very limited magnification levels.

Therefore, our ideal proposition for BreakHis classification reformulation would be: A multi-category classification system with a binary classification ability trained in a magnification-independent way. To our knowledge, this work is the first to present and analyze this reformulation and we called it ”MIM” which stands for magnification-independent multi-category classification. And as we discussed above, the ideal way to implement this reformulation would be a unique dual-stage model (a multi-category classification model with an integrated binary classification capability) trained with a magnification-independent approach.

3.3 Comparison between different approaches at pre- and post-processing phases

After finding the best reformulation for our BreakHis classification problem, we need to find the most adequate pre- and post-processing methods to fit this reformulation. In fact, BreakHis based CAD systems in the literature were presented with several approaches, different models and various trade-offs at pre and post-processing levels. Some works tried to compare and analyse the impact of different choices and various approaches on different classification tasks. However, their comparisons and analysis were biased towards their specific inner configurations with very restrictive case studies. To our knowledge no one of these attempts has elaborated a comparison between different works on BreakHis and analysed their results. Hence, we believe that an inter-works analysis with an objective view from above on all their adopted methods, models and best achieved results would be more reliable for efficient choices and well-founded assessments. In this section we will analyze and compare all these approaches, conclusions and results reported in BreakHis related works, with the aim to pick the most adequate pre and post processing solutions for our BreakHis classification problem reformulation deduced in the previous section. This ideal reformulation needs to be a unique multi-category classification model with a binary classification capability, trained in a magnification-specific approach.

Additionally, we will be taking into consideration BreakHis limitations, characteristics and experimentation protocol presented before. In addition to some of the generic methods that have never been used in BreakHis yet but could be very promising if explored in conjunction with our ideal reformulation to this problem.

3.3.1 Preprocessing methods comparison

For our ideal model hypothesis we will start with data preparation and necessary pre processing methods to fit our classification problem reformulation:

- **Data preparation** First of all, we need to elaborate an examination of suspicious noisy data samples with the help of a pathologist. Notably, to make sure if the malignant borderline patient (ID:13412) contains really features from ductal (DC) and lobular carcinoma (LC) sub-classes. Then, instead of eliminate this patient from the dataset as done in some related works, we will explore the possibility of separating both features parts into different images, and assign to each image its corresponding sub-class label. Thus, we could leverage the discriminative potential of this borderline case without causing a confusion to our multi-category classification model during its training.

  Secondly, we believe that for a fair comparison to other related works, we should follow the same unified experimentation protocol presented by BreakHis authors in [169] with the aim to generate 5 random folds of training(70%) /testing(30%) sets. However, unlike author's shared script[1] which generates these 5 folds for each magnification factor subset separately, ours will adopt a magnification-independent approach. Therefore, we will generate these 5 folds of training(70%) /testing(30%) once instead of four times, and using all magnification subsets combined (the whole dataset). To guarantee that our model generalizes well to unseen patients, we will be preserving the following constraint during our 5 folds creation: patients used to build the training set are not used in the test set.

- **Preprocessing methods** Several preprocessing methods were used in different BreakHis related works. Particularly, we will focus on the closest works to our reformulation. Notably, [50] which achieved the best results in both classification tasks combined. In fact, is fair to say that [50] is the state-of-the-art among works both classification tasks if considering that results achieved by [24] in multi category classification are not really reflecting this task it should be formulated, because authors in [24] evaluated it by taking in each test experiment one benign sub-class versus one malignant sub-class and not all the eight sub-classes at once. Moreover, it's true that [50] adopted a reformulation of both classifications with a magnification-specific approach which is not our case. However, its consideration as a close approach to ours is due to the fact that it adopts at patient level a kind of trainable ensemble learning method called MDT which considers classifiers of all magnification levels combined to make a final verdict to the departure of a given test patient. Hence, we could consider this information fusion of different magnification-specific factors to be a close reasoning to magnification-independent approach. Inspired by preprocessing methods adopted in these two frameworks [24,50], we will choose the following preprocessing methods:

  1. **Image resizing** Firstly, resize original images from 460×700 into 224×224 with the aim to fit the ResNet input layer (as we will see in next subsection the chosen CNN to be used will be ResNet).

[1] https://web.inf.ufpr.br/vri/databases/breast-cancer-histopathological-database-BreakHis/

2. **Stain normalization** Then apply a stain normalization to regularize color and brightness variance between different slides. However, instead of applying similar related work's stain normalization methods which requires a lot of expertise for target image choice (reference image), we believe that stainGAN [153] deep learning based method presented in section 5.3 could constitutes a perfect alternative to avoid the need of any target image choice or additional domain expert's expenses. In this case, we could simply use one of the well known datasets to be containing highly normalized histopathological slides as a target domain and train StainGAN model to transfer staining style from this dataset to BreakHis images. For instance, as already proven in [153], MITOS-ATYPIA-14 challenge[2] dataset seems to be a good ground truth for StainGAN training. In fact, this dataset contains breast cancer histopathological images taken with two different scanners. Thus, we could choose one scanner subset and use it as a target domain, while using BreakHis images as an input domain for StainGAN model. This experiment will results in a normalized BreakHis version with similar staining style to one of MITOS-ATYPIA-14 scanner subsets.
3. **Data augmentation** Afterwards, giving all comparison studies elaborated in [9, 50, 128, 174, 193] proving the huge improvement at image and patient levels brought by data augmentation especially with deep learning models, we choose to consider a data augmentation approach, and regarding data augmenter, we will explore a composition of common techniques in all best models and the closest ones to our reformulation [50, 119], which are: random rotation and random flipping. Furthermore, we prefer to apply these data augmentation with a data balancing constraint between all sub-classes, in order to address the uneven data distribution between them as done in the multi-category models proposed in [50].

Regarding the chosen order of the proposed preprocessing methods above, image resizing was placed intentionally as a first step, because when stain normalization or other preprocessing methods are applied on prior downsized images we could considerably gain in preprocessing time required to apply these methods on higher resolution version.

3.3.2 DL models and training strategies comparisons

After choosing the most suitable preprocessing methods for our ideal model hypothesis, we will pick the necessary DL model and training strategy to fit our reformulation for BreakHis classification.

To classify BreakHis images, each related work reported in this paper adopted one of the four following approaches:

- **Handcrafted features extractor + Traditional classifier** This first approach is a purely traditional one, where one or many handcrafted features descriptors are used to extract from BreakHis images their inner features. Then, these features with their corresponding labels are provided to a traditional classifier (a non-deep learning model) for training and classification.

- **Handcrafted features extractor + deep learning model** This second approach is similar to the first one, except the fact that it uses a deep learning model for training and classification in the second stage instead of a traditional classifier.

- **Deep learning model as features extractor + Traditional classifier** This third approach is similar to the first one, except the fact that it uses a deep learning model for features extraction

[2] https://mitos-atypia-14.grand-challenge.org/dataset/

instead of any traditional handcrafted features descriptor. Generally, in this case a pretrained CNN is used for features extraction from its last layers.

- **Global end-to-end deep learning model** This fourth approach is a purely deep learning one, where a deep learning model is provided and trained in an end-to-end manner with BreakHis images and their corresponding labels.

As expected all best results in all classification tasks were achieved when global deep learning CNNs were trained in an end to end manner without prior handcrafted features extractors or traditional post classifiers. Therefore, we believe that a global CNN trained in an en-to-end manner would be more adequate for our ideal model hypothesis. In addition to the outstanding results achieved by this approach in several related works, from a theoretical point of view this choice could be justified by the conformity of CNNs architectures to our magnification-independent reformulation. In fact, our reformulation for this classification problem consists of a model trained with all magnification levels images combined and as demonstrated in [63], the first CNN layers learns low magnification features while the last ones captures the higher level features. Thus, a CNN trained with this hierarchical learning process throughout its multiple layers is the most suitable model for the nature of histopathological images especially when adopting a magnification-independent approach. Regarding the CNN architecture choice, most BreakHis related works with deeper CNNs outperformed the shallower based ones in all classification tasks. Particularly, ResNet and all its variants especially those adopted in [50, 119] which are clearly outperforming all other CNNs in the binary and multi-category classification tasks with both magnification-specific or magnification-independent approaches. For our ideal model hypothesis, we prefer to explore ResNet CNN model that achieves very high results in most reformulations.

After picking the most adequate preprocessing and CNN model for our reformulation of BreakHis classification problem, we need to decide which training strategies are the suitable for this model to explore in future works. In the following points we will discuss the necessary training and learning approaches to be adopted for our ideal model hypothesis:

- **Patch based vs whole slide based training** As we mentioned before our proposed ResNet will be trained in a supervised end-to-end manner provided with images and their corresponding labels. To feed this CNN we could use one of the following approaches:

  - A whole slide based approach where the CNN model is provided in the input layer with WSIs after applying all necessary preprocessing operations
  - A patch based approach where the CNN model is provided with small size patches extracted from the WSIs after applying all necessary preprocessing operations

  Mostly, all related works on BreakHis dataset [9,63,67,119,121,122,127,174,193] that achieved a higher accuracy at image and patient levels in all reformulations adopted a whole slide approach instead of a patch based one, except [50] which extracted for each class or sub-class a different number of overlapping patches with size 224 × 224 using a sliding window approach. Nevertheless, images in [50] were resized to 341 × 224 before 224 × 224 patches extraction. By consequences, roughly speaking we could consider this large-size patch extraction step adopted in [50] as a whole slide approach, because resulted patches are having almost similar size to source images which by their turn are similar to original images because they were only downsized during the preprocessing and still contains the same original information. In fact, we believe that the key point of results in [50] is the implicit data balancing operation which consists of taking into consideration the imbalance ratio of each class to decide how many patches to be extracted from it. Therefore, we prefer to opt for a whole slide based approach to train our proposed ideal model and as we mentioned before in the preprocessing section original WSIs will be downsized from 460×700 into 224×224 in order to fit the ResNet input layer. This WSI training based approach could help us also avoid any additional domain expert's expenses for patches labeling, because when a patch based approach is adopted extracted patches are not obligatory having the same labels as their source images and needs to be annotated, which pushed some authors to opt for a MIL approach or a deep active learning method.

- **Features distance constraint** To elaborate a multi-category classification model with a very high discriminative capability between different sub-classes, we believe that is necessary to integrate a learning constraint on this model during its training as shown in [67, 193] to separate different classes as wider as possible in their features space. In fact, this choice is justified by the complexity of this task in comparison to the standard binary classification one, with subtle visual and morphological appearance differences between the eight sub-classes as depicted in [9, 67]. This learning constraint applied on features space distance between intra-class and inter-class training instances is necessary for controlling features similarities of different sub-classes and improve the model ability to discriminate each one in the features space correctly. Regarding the learning constraint's choice, we prefer to use the one presented in [67], simply because the latter's distance constraint is well defined and explained and could be directly injected to the loss function during the training, unlike the one used in [193] which is unclear and not explicitly formulated.

- **Transfer learning** Naturally, when adopting a CNN for a relatively small dataset as our case with various classes we should adopt a transfer learning approach. Most CNN based works with BreakHis images and especially those with an end to end global CNN as our proposition, adopted a transfer learning approach by fine tuning an ImageNet pre-trained CNN on BreakHis classification task. Results of which guaranteed to be a very good trade-off between model complexity and results accuracy, in addition to providing a good initial state for the adopted CNN's weights. As demonstrated in [204], when adopting any transfer learning approach we should be aware of the gap between ImageNet and BreakHis domains. Therefore, an adaptation should be elaborated for the used pre-trained CNN. To achieve this adaptation we choose to explore the interesting dual-stage fine-tuning approach presented in [210]. The pre-trained CNN used in the latter, undergone a two stage fine tuning method, starting by freezing the first convolutions while retraining only the fully connected layers, then fine tuning all layers. This choice is justified by two factors:
  - On one hand, as explored in [210] this combined approach achieved higher results than approaches where only one of these fine-tuning stages is performed.
  - On the other hand, we believe that when following the same order of both stages we could build a good adaptation to mitigate the knowledge gap between the two different domains (ImageNet and BreakHis). In fact, starting with retraining the last layers only present a good initial adaptation step, because ImageNet specific features which are the less common features with BreakHis images are mostly hidden in these last layers. Then, retraining the whole network in a second step seems to be adequate for further adaptation, because as depicted in [63] low and mid-range layers are containing the majority of BreakHis specific features.

- **Ensemble learning** as shown in [9, 36, 50, 61, 168, 169, 174, 193, 210] ensemble learning is necessary to achieve better results in terms of accuracy and stability over all magnification test sets. Therefore, adopting an ensemble could help us improve the overall performance of our proposed model. Regarding the ensemble learning strategy to use, we propose to explore an ensemble of all RestNet variants which achieved higher results in BreakHis classification task (i.e ResNet-50 ResNet-101 and ResNet-152). Regarding the aggregation rule between these different classifiers, the trainable MDT approach in [50] seems to be a good choice to achieve higher accuracy, but cannot be adapted to our proposed model, because as presented before this MDT method in opposition to our magnification-independent reformulation requires images at different magnification levels. Thus, we could explore the efficiency of various fusion rules starting with the max rule which outperformed the sum and product rules when compared with them in [168], we could also evaluate the majority voting rule used in [61], or even the average rule as used in [174] with ResNet CNNs.

Furthermore, we believe that we should adopt the same metrics as the one used by the majority of BreakHis related works and authors of this dataset themselves: Image Level Accuracy (ILA) in order to evaluate our ideal model hypothesis and guarantee a fair comparison to other related works. Moreover, and to allow a possible comparison with all other reformulations for this problem, the ideal model should be evaluated in all classification tasks including the binary, malignant multi-category and benign multi-category classifications.

3.4 Summary and ideal model

- **Summary** After presenting lessons learnt from all these works, we discussed and analysed their various approaches and findings. Our analysis of all DL models, pre- and post-processing methods and the comparison between the four different reformulations present in the literature, led us to conclude the following points:

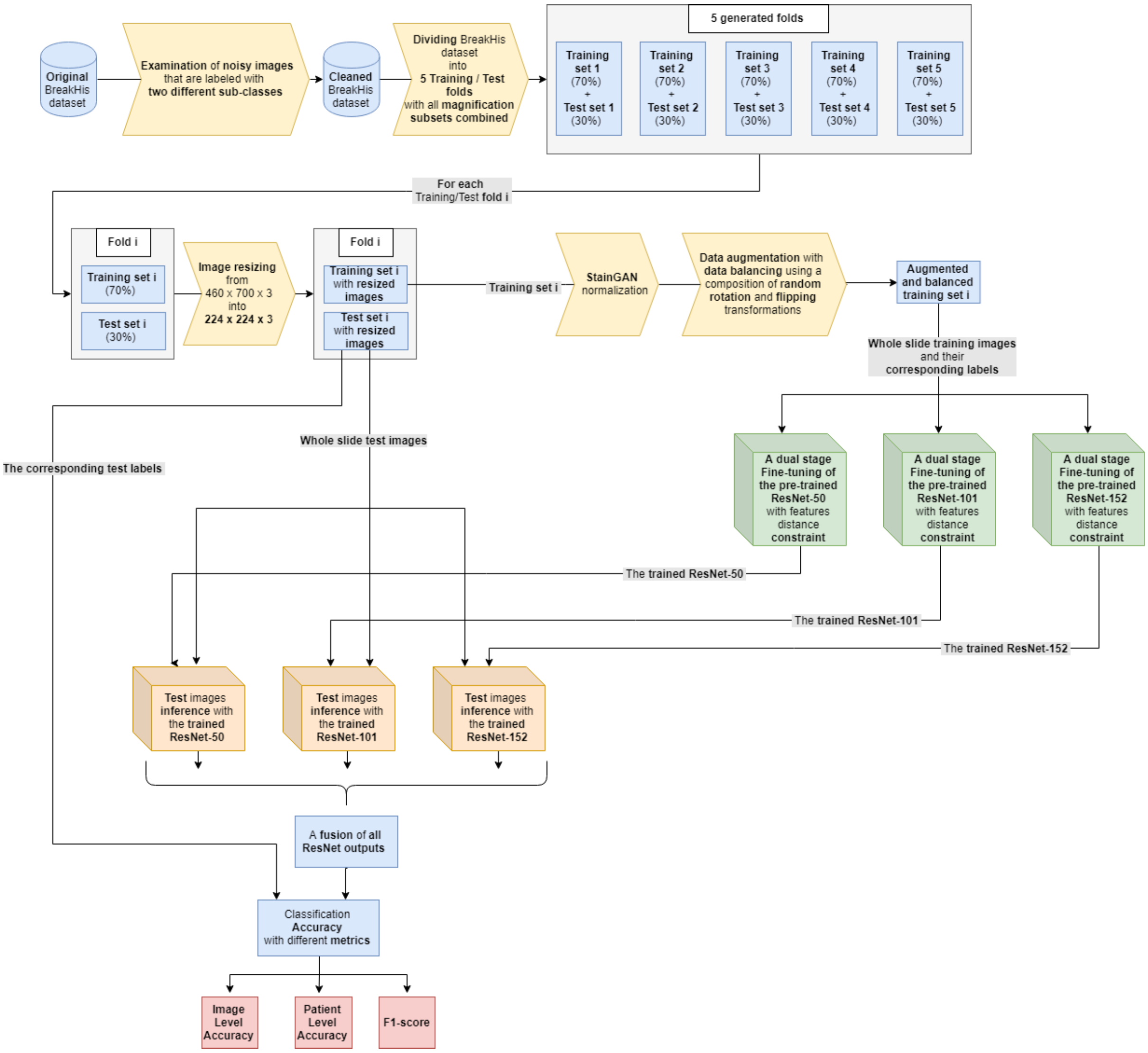
  - A multi-category classification system with a binary classification ability trained in a magnification-independent way would be the most adequate reformulation for BreakHis problem form a practical as well as a clinical standpoint.

  In the second stage, we elaborated a comparison between all pre and post-processing approaches reported in the literature with more focus on deep learning based ones in order to find out what would be the most suitable methods, models, and learning approaches to fit our concluded best reformulation for this problem. After these comparisons, our proposed hypothesises to build an ideal model for this problem are the following:

  - Deep learning models would be more adequate for this problem than traditional handcrafted based models.
  - Among all used deep learning models, CNNs would be more adequate for this problem.
  - Among all used CNN models, a ResNet trained in an end-to-end manner without prior features extractor or post-classifier would be the ideal choice for this problem.
  - A transfer learning approach would be of high interest for training a CNN with this small dataset.
  - As a transfer learning approach with a pre-trained CNN, fine-tuning use would be better than features extraction use.
  - For fine-tuning, it would be better to adopt a dual-stage fine-tuning, by firstly retraining the last layers only before retraining the whole network in a second stage.
  - Providing the used CNN with whole slide images(WSIs) would be better than adopting a patch-based approach.

  - Applying a constraint on features distance such as the one used in [67] during the CNN training could help us ease the similarity issue between different sub-categories.
  - Ensemble learning would be of high interest in this problem.
  - As an ensemble learning approach, we could merge decisions made by all ResNet variants that proved to be the more efficients when used alone in this task (ResNet-50, ResNet-101 and ResNet-152).
  - As preprocessing methods, the ideal combination would be image resizing, stain normalization and data augmentation.
  - For image resizing, it would be better to resize the images to fit the CNN's input layer.
  - For stain normalization, a deep learning GAN based method such as StainGAN would be a better choice than classical methods.
  - For data augmentation, a promising combination would be random rotation with random flipping.
  - To tackle data imbalance issue, data augmentation would be better applied with a data balancing purpose between different sub-categories.

- **Ideal model** These comparisons and their conclusions when merged together allowed us to formulate a global hypothesis. This hypothesis is a proposition of what would be an ideal model to build a CAD system for BreakHis or even for similar breast cancer datasets. A summary of the proposed ideal model with the adopted DL model, data pre- and post-processing approaches is presented in the figure 3.1.



Original BreakHis dataset
Examination of noisy images that are labeled with two different sub-classes
Cleaned BreakHis dataset
Dividing BreakHis dataset into 5 Training / Test folds with all magnification subsets combined
5 generated folds
Training set 1 (70%) + Test set 1 (30%)
Training set 2 (70%) + Test set 2 (30%)
Training set 3 (70%) + Test set 3 (30%)
Training set 4 (70%) + Test set 4 (30%)
Training set 5 (70%) + Test set 5 (30%)
For each Training/Test fold i
Fold i
Training set i (70%)
Test set i (30%)
Image resizing from 460 x 700 x 3 into 224 x 224 x 3
Fold i
Training set i with resized images
Test set i with resized images
Training set i
StainGAN normalization
Data augmentation with data balancing using a composition of random rotation and flipping transformations
Augmented and balanced training set i
Whole slide training images and their corresponding labels
A dual stage Fine-tuning of the pre-trained ResNet-50 with features distance constraint
A dual stage Fine-tuning of the pre-trained ResNet-101 with features distance constraint
A dual stage Fine-tuning of the pre-trained ResNet-152 with features distance constraint
The corresponding test labels
Whole slide test images
The trained ResNet-50
The trained ResNet-101
The trained ResNet-152
Test images inference with the trained ResNet-50
Test images inference with the trained ResNet-101
Test images inference with the trained ResNet-152
A fusion of all ResNet outputs
Classification Accuracy with different metrics
Image Level Accuracy
Patient Level Accuracy
F1-score

Figure 3.1: The proposed framework for the ideal model hypothesis

3.5 Ideal model evaluation and learnt lessons

To our knowledge, our work is the first one in the literature to analyze the magnification-independent multi-category (MIM) classification on BreakHis dataset. In this section, we evaluate experimentally the designed ideal model for this task and report the learnt lessons. The structure of this experimental study is presented as follows:

- Section 3.5.1 presents the used experimental protocol, then report the achieved results
- Section 3.5.2 summarizes the lessons learnt from this experimental study

3.5.1 Experimental results

In this part we explored the proposed ideal MIM approach using DL. In addition, we analyzed the impact brought by data augmentation and stain normalization as preprocessing techniques and fine-

Classification task	**Fine-tuning stage**	**The ideal model** **Without DA and SN**	**The ideal model** **With DA(Rot,Flip)**	**The ideal model** **With DA(Rot,Flip) and SN(Macenko)**	**The ideal model**
Binary	last layers	78.1 ± 2.5	80.3 ± 3.0	75.0 ± 1.3	76.3 ± 0.7
	last layers then all layers	83.5 ± 1.5	88.1 ± 2.1	87.2 ± 1.4	**88.9 ± 2.5**
Malignant multi-category	last layers	62.3 ± 1.5	61.2 ± 2.0	60.1 ± 1.2	60.31.3
	last layers then all layers	57.4 ± 2.0	60.1 ± 1.7	56.0 ± 2.3	**63.6 ± 2.2**
Benign Multi-category	last layers	33.8 ± 1.9	38.4 ± 2.3	35.0 ± 2.4	37.2 ± 1.5
	last layers then all layers	47.6 ± 1.6	**52.7 ± 2.3**	39.0 ± 1.5	41.3 ± 0.9

Table 3.1: Results of each task of the proposed MIM ideal model at ILA.

tuning different layers of the used CNN. We believe that this formulation reflects more the pathologists workflow that analyzes the exact sub-category to decide the corresponding treatment for their patients. To train our model we used a magnification-independent approach, and for each one we used five trials with different folds, where each fold consists of 70% of BreakHis images as a training set and 30% for test. For each trial, we followed the major constraint adopted by BreakHis authors in MSB which guarantee that patients who were used for training were not reused during the test phase. This proposed model has been designed based on comparison and analysis of different works reported in the literature. In fact, this experimental study will evaluate the efficiency of the chosen DL models, the proposed pre- and post-processing methods and explore their coherency with the suggested ideal reformulation for this problem.

Recall that our MIM system is able to perform a binary classification (benign or malignant) as well as a multi-category classification for each one of these two main malignancy classes. Thus, we evaluated our proposed ideal model results according to each one of these three classification tasks and reported their performance in Table 3.1. Results are presented according to the classification accuracy mean value and standard deviation at image level (ILA) over five trials. These results are also organized at the preprocessing level to explore the improvement brought by stain normalization and data augmentation methods. Furthermore, we present the performance of our model at each stage of the adopted fine-tuning strategy.

In general, we can see that results of the binary classification task are very competitive when compared to those reported within the MIB reformulation in table 2.5, especially those using the same data partition protocol (70%,30%) [14, 61]. However, results for the MIM tasks achieved low accuracy in the malignant sub-classes classification and even lower in the benign sub-classes classification. In addition, at the binary classification and malignant multi-category classification tasks, our proposed data augmentation and stain normalization approaches achieved better results than other standard data augmentation or stain normalization methods. On the other side, we observed that in the majority of cases adopting a two-stages fine tuning approach has been always better than fine tuning only the last layers of the adopted CNN.

3.5.2 learnt lessons

As it can be seen from this experimental study, the MIM classification accuracy is very low in comparison to the MSM counterpart. This could be explained by the following reasons:

- Learning the differences between the eight subcategories regardless of their magnification levels is much harder for the CNN, especially for the benign sub-categories (minority sub-classes) even after data balancing.
- The irregularities generated by the optical microscopic magnifications used to collect BreakHis data. When a ROI in a breast tissue is magnified optically, some new morphological components appears as long as we dive deeper into the tissue. In other words, it remains very hard for a CNN

to learn the most representative features when the same sub-class contains images at different levels with different morphological structures.

Given the current BreakHis structure and characteristics, the most reasonable approach is MSM as it maintains the same clinical value as MIM, while it scarifies only its practical counterpart. Otherwise, to achieve good results with the MIM approach itself, several solutions must be explored such as data fusion at the same magnification levels between BreakHis and other available histopathological breast cancer datasets. However, since these datasets have different labeling purposes, this data fusion needs more adaptation efforts especially in terms of expert annotation.

3.6 Conclusion

The proposed model in this chapter is the product of the comparison and analysis of different BreakHis related works in terms of the adopted reformulation and all used classification techniques. From a practical and medical point of view, our study concluded that in a real clinical routine, a CAD system based on the MIM reformulation is the best approach for this problem. To establish the ideal model for this problem that will correspond to the MIM reformulation, we merged the best performing DL approach at both pre- and post-processing. Our expectations were that such a sophisticated combination will outperform the state-of-the-art results in this problem. However, after the experimental evaluation of this hypothesis, we wind up with a limited accuracy for the three classification tasks in this problem. The most reasonable explanation for this outcome is that given the current state and organisation of BreakHis dataset, building such an automatic system using the best CNN configuration does not lead to an acceptable accuracy yet. For instance, we could consider that MSM is the closest reformulation to MIM from a clinical standpoint that fits the actual BreakHis composition. Whereas, for an MIM based model, more labeling effort are still required and would be of high interest if established on similar datasets such as Bioimaging and MITOSATYPIA with the aim to meet BreakHis structure under a data fusion scenario. This point brings us back to the main hypothesis of this thesis and make us wonder either a well designed DL model is immune to any data related issue and can perform efficiently in every scenario. Here, in the case of BreakHis dataset, we can see clearly that even the best combination of DL at pre- and post-processing phases can not be as good as we expect for a new problem that is different than the dataset main objective. Thus, we can conclude from this analysis that the dataset quality remains the most important key in allowing a DL model to either perform well or not. Another conclusion that we deduced from this study is: the fact that BreakHis has been mainly built for MSB classification constitutes a limitation for us when we want to use this dataset for other classification tasks. In fact, this observation comes from the good performance achieved by BreakHis in MSB classification with many DL models and preprocessing techniques. This point can be reformulated as follow: Besides its data quality, BreakHis data structure and the goal that it was originally built for, affect DL models performance when trying to address a new classification problem. To overcome these limitations, in the next chapter of this thesis, we will create our own dataset where we have control on the annotation quality and the classification structure and make sure that it would be suitable to be used with DL models. The perfect scenario in this thesis would have been to create our own breast cancer dataset but this task is very difficult and requires a huge medical expertise as it is the case in the creation of any other biomedical images dataset. Therefore, the dataset we will create will also contain images, but this time we will be tackling another complex application of DL which is satellite images classification; which will allow us to continue the evaluation and analysis of our hypothesis but in a new different scenario.

CHAPTER 4

Sentinel2GlobalLULC: A deep-learning-ready Sentinel-2 RGB image dataset for global land use/cover mapping

Contents

This chapter is submitted as a manuscript under review in Scientific Data-Nature Publishing Group , and published as a preprint in BioRxiv [16].

4.1 Introduction and motivation

Let us first give a detailed introduction to the main topic of the present chapter: Land-Use and Land-Cover (LULC) mapping is relevant for many applications, from climate modelling to territorial, agricultural and urban planning. Global LULC products are continuously developing as remote sensing data and related methods grow. However, there is still a very low consistency among LULC products due to low accuracy for some regions of the world and several LULC types. In this chapter, we introduce Sentinel2GlobalLULC, a Sentinel-2 RGB image dataset, built from the consensus of 15 global LULC maps available in Google Earth Engine (GEE). Sentinel2GlobalLULC v1.1 contains 195572 RGB images organized into 29 global LULC mapping classes. Each image has 224 × 224 pixels at 10 × 10 m spatial resolution each, and was built as a cloud-free composite from all Sentinel-2 images acquired between June 2015 and October 2020. Each image has a unique LULC type annotation, a level of consensus and its name contains the reverse geo-referencing information, and the corresponding global human modification index. Sentinel2GlobalLULC was optimized to be used with the state-of-the-art DL models with the aim of building precise and robust global or regional LULC maps. The creation of this dataset as we announced in the conclusion of the last chapter will help us to answer the question related to either the data structure and the classification task that it was originally built for, affect also DL models performance or not. In this new setting we will have a total control on the annotation quality and the classification structure to ensure that this dataset will be suitable to be used with DL models to solve this specific problem which is LULC mapping. In this chapter, we introduce Sentinel2GlobalLULC dataset, explain the used methodology to built it in details and report the results of its technical validation. Sentinel2GlobalLULC dataset and its corresponding metadata are stored in the following Zenodo repository(DOI:10.5281/zenodo.5055632).

4.2 Background & Summary

Land-Use and Land-Cover mapping aims to comprise the continuous biophysical properties of the Earth surface into synthetic categorical classes of natural or human origin, such as forests, shrublands, grasslands, marshlands, croplands, urban areas or water bodies [40]. High resolution LULC mapping plays a key role in many fields, from natural resources monitoring, to biodiversity conservation, urban planning, agricultural management or climate and earth system modelling [115, 140, 190]. Multiple LULC products have been derived using satellite information at the global scale (Table 4.2), contributing to a better monitoring and understanding of our planet [37, 139].

However, despite the acceptable accuracy of each individual product, a considerable disagreement between products has been reported [11,29,74,105,114,118,142,156,173,183–185,190,195,197,208,211]. There are several methodological reasons behind this problem:

- Different satellite sensors with different spatial resolutions were used in each product, so the difference in precision from coarse to fine resolution partially determines the final quality of each product.
- Different preprocessing techniques, like atmospheric corrections, cloud removal and image composition were used in each LULC product.
- Each LULC product has a different temporal updating rate, some are regularly updated, whereas others have never been updated.
- Different classification techniques, field-data collection approaches and subjective interpretations were used to create each product.
- Different classification systems (LULC legends) were adopted in each product, usually focused on distinct applications.
- Different validation techniques and different ground truth reference data were used in each product, which impedes a reliable accuracy comparison.

Over the last few years, several attempts have been made to overcome these inconsistencies with a harmonised approach capable of providing greater control in the validation and comparison over the growing number of existing LULC products [51, 104]. Even though, users still have some issues regarding appropriate product selection due to the following factors:

- In most cases, users are unable to find a product that fits their desired LULC class or geographic region of interest [54, 198].
- These products are usually collected at a coarse resolution, which makes analysis at a finer scale difficult [185].
- These products offers a limited number of LULC classes that usually change from one product to another [48].

In parallel, Deep artificial neural networks, also known as Deep learning (DL), are increasingly used in LULC mapping with promising potential [214]. This interest is motivated by the good performance of DL models in computer vision and, particularly of Convolutional Neural Networks (CNNs) in remote sensing image classification and many applications [15, 107, 131, 143, 159]. However, to reach high performance, DL models need to be trained on large smart datasets [205]. The concept of smart data involves all preprocessing methods that improve value and veracity of the data in addition to the quality of the associated expert annotations [106].

Dataset	**Source**	**Source mapping type**	**Number of images**	**Image Size**	**Spatial Resolution**	**No. Bands**	**No. Classes**	**Extent**
ISPRS Vaihingen ([146])	-	Airborne	33 im	2000 x 2000	0.09	3	6	Local
ISPRS Postdam ([146])	-	Airborne	38 im	6000 x 6000	0.09	3	6	Local
Brazilian coffee scenes ([135])	SPOT-5	Spaceborne	50,004 im	64 x 64	10	3	3	Local
SAT-4 ([13])	NAIP program	Airborne	500,000 im	28 x 28	1	4	4	Local
SAT-6 ([13])	NAIP program	Airborne	405,000 im	28 x 28	1	4	6	Local
UCMerced ([200])	OPLS	Airborne	2100 im	256 x 256	0.3	4	21	Local
Zeebruges (link)	LiDAR	Airborne	100,000 im	10 x 10	0.05	3	8	Local
WHU-RS19 ([34])	Google Earth	Airborne	1005 im	600 x 600	Up to 0.5	3	19	Local
SIRI-WHU ([207])	Google Earth	Airborne	2.240 im	200 x 200	2	3	12	Local
RSSCN7 ([215])	Google Earth	Airborne	2800 im	400 x 400	-	3	7	Local
RSC11 (link)	Google Earth	Airborne	1232 im	512 x 512	0.2	3	11	Local
NWPU-RESISC45 ([29])	-	-	31,500 im	256 x 256	3̃0-0.2	3	45	Local
AID ([196])	Google Earth	Airborne	10,000 im	600 x 600	8̃-0.5	3	30	Local
BigEarthNet ([173])	Sentinel-2	Satellite	590,326 img.	-	-	-	-	10 European countries
SpaceNet-7 ([187])	Dove Satellite Constellation Planet Labs'	Satellite	img.	-	-	-	-	100 cities

Table 4.1: List of existing Land-Use and Land-Cover (LULC) datasets ready for training Deep Learning (DL) models.

Currently, there exist several remote sensing datasets derived from satellite and aerial imagery ready for training DL models for LULC mapping (Table 4.1). However, they still suffer from some limitations, particularly to be used with DL models:

- None of them represent the global heterogeneity of the broad categories of LULC classes throughout the Earth. Usually, they are biased towards specific regions of the world, limited to national or continental scales, which can propagate such bias to the DL models [56, 126, 199]. As illustration, the reader can see how visual features of urban areas may change from one country to another (Figure 4.1).
- They are relatively small and have only hundreds to few thousands of annotated data records [73].
- They suffer from high variability in atmospheric conditions, and they have high inter-class similarity and intra-class variability, which makes class differentiation difficult [73, 73].

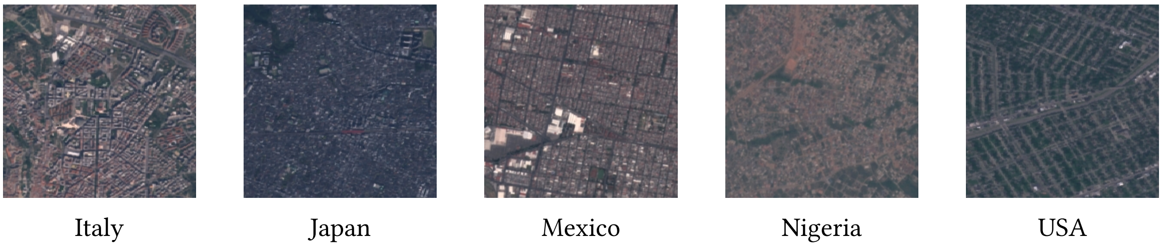

Italy Japan Mexico Nigeria USA

Figure 4.1: Illustration from different countries of the Sentinel-2 satellite images corresponding to one of the 29 Land-Use and Land-Cover (LULC) classes (e.g. Urban and built-up area) extracted from Sentinel2GlobalLULC dataset. Each image has 224 × 224 pixels of 10 × 10 m resolution. Pixel values were calculated as the 25th-percentile of all images captured between June 2015 and October 2020 that were not tagged as cloudy. Fifteen LULC products available in GEE agreed in annotating each image to represent one LULC class

To overcome these limitations, we introduce in this thesis Sentinel2GlobalLULC, a smart dataset with 29 fully annotated LULC classes at global scale built with Sentinel-2 RGB imagery. Every image in this dataset is geo-referenced and labeled with its corresponding LULC annotation. Each image label was carefully built from a consensus approach of up to 15 global LULC maps available on GEE [59]. We released a Tagged Image File (tif) and jpeg version of each image. Moreover, we attached to these images, a Comma-separated values (CSV) file for each LULC class containing the coordinates of each image center, and additional metadata. Sentinel2GlobalLULC could be used to train and/or evaluate DL based models for global LULC mapping. Sentinel2GlobalLULC aims to foster the creation of accurate global LULC products exploiting the advantages that currently offer deep learning models. We expect this dataset to improve our understanding and modelling of natural and human systems around the world.

4.3 Methods

To build Sentinel2GlobalLULC, we followed two main steps. First, we established a spatial consensus between 15 global LULC products for 29 LULC classes. Then, for each class, we carefully extracted the maximum possible number of Sentinel-2 RGB images in 224 × 224 pixel tiles at 10 m/pixel spatial resolution. Both tasks were implemented using GEE, an efficient programming, processing and visualisation platform that allowed us to have free manipulation and access to all used LULC products and Sentinel-2 imagery, simultaneously.

4.3.1 Finding spatio-temporal agreement across 15 global LULC products

To establish the spatio-temporal consensus between different LULC products for each one of the 29 LULC classes, we followed four steps: 1) identification of the LULC products to use for the consensus, 2) standardization and harmonization of the LULC legend that was subsequently used as annotation, 3) spatio-temporal aggregation across selected LULC products, and 4) spatial reprojection and tile selection based on optimized spatial purity thresholds.

4.3.1.1 Global LULC products selection

To find areas of high consensus in their LULC mapping, we selected the 15 global LULC products available in GEE (Table 4.2). Reaching consensus across such rich diversity of LULC products, in terms of spatial resolution, time coverage, satellite source, LULC classes and accuracy, made our LULC annotation robust. This way, each image was annotated with a LULC class only if all combined products agreed (i.e., 100% of agreement in space and time). For some LULC classes, we had to decrease the purity threshold to reach a large number of samples. The purity level is always provided as metadata for each image (details in the subsection Re-projection and Selection of purity threshold).

LULC product	Satellite or Spaceborne	Resolution	Used years	Reference
P1: MCD12Q1.006 MODIS LULC Type Yearly Global 500m LULC Type1: Annual International Geosphere-Biosphere Programme (IGBP) classification (version 6)	Aqua, Terra	500 meters	2017 to 2019	[172]
P2: MCD12Q1.006 MODIS LULC Type Yearly Global 500m LULC Type 2: Annual University of Maryland (UMD) classification (version 6)	Aqua, Terra	500 meters	2017 to 2019	[172]
P3: MCD12Q1.006 MODIS LULC Type Yearly Global 500m LULC Type 3: Annual Leaf Area Index (LAI) classification (version 6)	Aqua, Terra	500 meters	2017 to 2019	[172]
P4: MCD12Q1.006 MODIS LULC Type Yearly Global 500m LULC Type 4: Annual BIOME-Biogeochemical Cycles (BGC) classification (version 6)	Aqua, Terra	500 meters	2017 to 2019	[172]
P5: MCD12Q1.006 MODIS LULC Type Yearly Global 500m LULC Type 5: Annual Plant Functional Types classification (version 6)	Aqua, Terra	500 meters	2017 to 2019	[172]
P6: Copernicus Global LULC Layers: CGLS-LC100 collection 3 (version 3.0.1)	PROBA-V	100 meters	2017 to 2019	[22]
P7: Global Forest Cover Change (GFCC) Tree Cover Multi-Year Global 30m (version 3.0)	Multi-satellite	30 meters	2015	[152]
P8: GlobCover: Global LULC Map (version 2.0)	ENVISAT	300 meters	2009	ESA 2010 and UCLouvain
P9: GFSAD1000: Cropland Extent 1km Multi-Study Crop Mask, Global Food-Support Analysis Data (version 0.1)	Multi-satellite	1000 meters	2010	[179]
P10: Global PALSAR-2/PALSAR Forest/Non-Forest Map (version fnf)	ALOS, ALOS 2	25 meters	2017	[158]
P11: Hansen Global Forest Change (version 1.7)	Landsat 8	1 arc seconds	2000 to 2019	[68]
P12: Global Forest Canopy Height (version 2005)	Lidar	30 arc seconds	2005	[160]
P13: JRC Yearly Water Classification History (version 1.2)	Landsat (5,7,8)	30 meters	2017 to 2019	[134]
P14: JRC Global Surface Water Mapping Layers (version 1.2)	Landsat(5,7,8)	30 meters	1984 to 2019	[134]
P15: Tsinghua FROM-GLC year of change to impervious surface(version 10)	Landsat	30 meters	1985 to 2019	[57]

Table 4.2: Main characteristics of the 15 global Land-Use and Land-Cover (LULC) products available in GEE that were combined to find consensus in the global distribution of 29 main LULC classes

4.3.1.2 Standardization and Harmonization of LULC legends

Land cover (LC) data describes the main type of natural ecosystem that occupies an area; either by vegetation types such as shrublands, grasslands and forests, or by other biophysical classes such as permanent snow, bare land and water bodies. Land use (LU) includes the way in which people modify or exploit an area, such as in urban areas or agricultural fields.

To build our 29 LULC classes nomenclature, we established a standardization and harmonization approach based on expert knowledge. During this process we took into account the needs of different practitioners in the LULC mapping field and the thematic resolution of the global LULC legends available in GEE. Hence, our nomenclature consists of 23 LC and 6 LU distinct classes interoperable through a set of criteria across 15 LULC products specified in our consensus rules (Table 4.3). A six-level (L0 to L5) hierarchical structure was adopted in the creation of these 29 LULC classes (Figure 4.2).

The LC part contains 20 terrestrial ecosystems and three aquatic ecosystems. The terrestrial systems are: Barren lands, Grasslands, Permanent snow, Moss and Lichen lands, Close Shrublands, Open Shrublands, in addition to 12 Forests classes that differed in their tree cover, phenology, and leaf type. The aquatic classes are: Marine water bodies, Continental water bodies, and Wetlands; furthermore, wetlands are divided into three classes: Marshlands, Mangroves and Swamps. The LU part is composed of urban areas and five coarse cropland types that differed in their irrigation regime and leaf type.

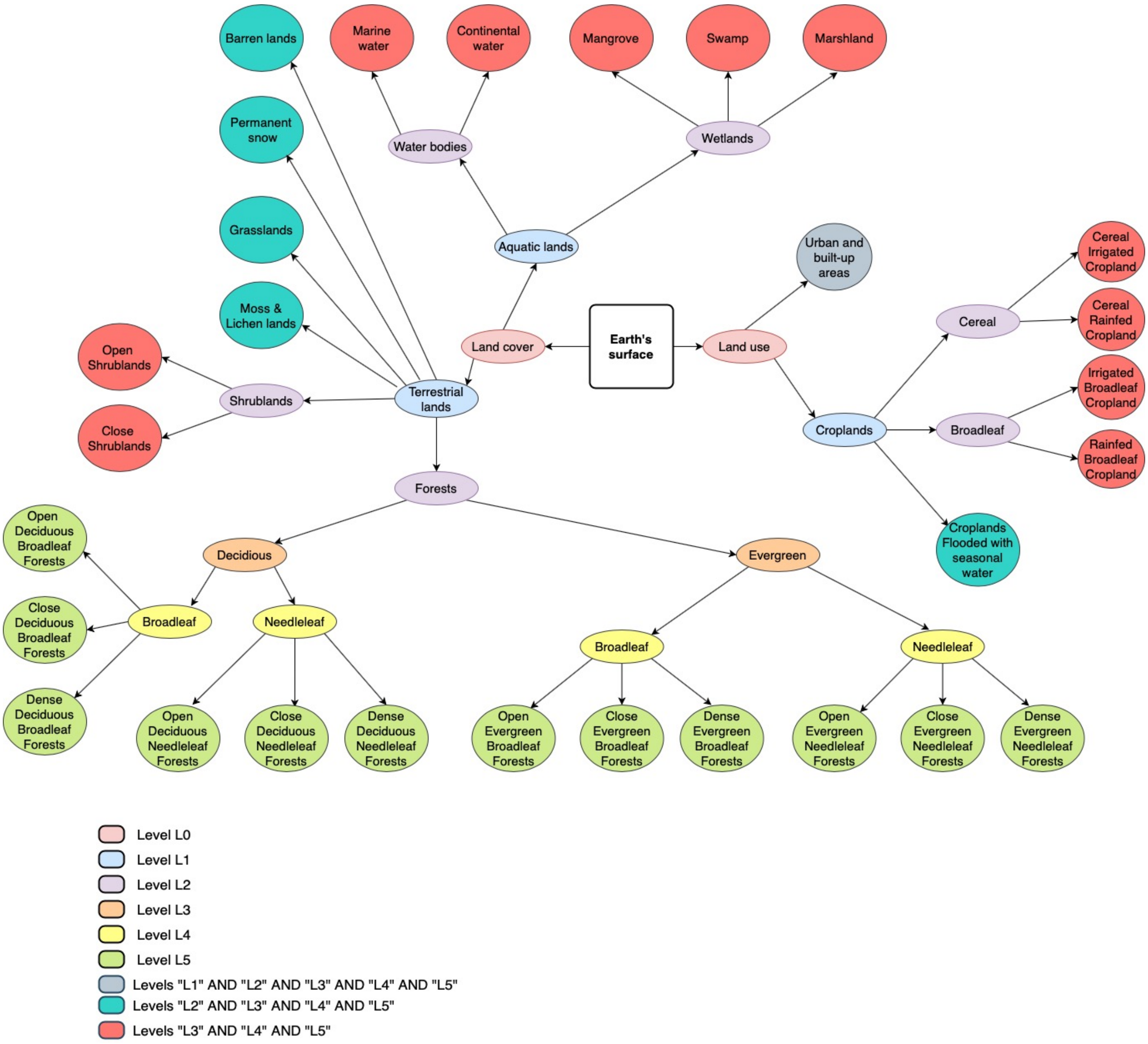

Barren lands
Marine water
Continental water
Mangrove
Swamp
Marshland
Permanent snow
Water bodies
Wetlands
Grasslands
Aquatic lands
Urban and built-up areas
Cereal Irrigated Cropland
Moss & Lichen lands
Cereal
Cereal Rainfed Cropland
Open Shrublands
Land cover
Earth's surface
Land use
Irrigated Broadleaf Cropland
Shrublands
Terrestrial lands
Close Shrublands
Croplands
Broadleaf
Rainfed Broadleaf Cropland
Forests
Open Deciduous Broadleaf Forests
Decidious
Evergreen
Croplands Flooded with seasonal water
Close Deciduous Broadleaf Forests
Broadleaf
Needleleaf
Broadleaf
Needleleaf
Dense Deciduous Broadleaf Forests
Open Deciduous Needleleaf Forests
Close Deciduous Needleleaf Forests
Dense Deciduous Needleleaf Forests
Open Evergreen Broadleaf Forests
Close Evergreen Broadleaf Forests
Dense Evergreen Broadleaf Forests
Open Evergreen Needleleaf Forests
Close Evergreen Needleleaf Forests
Dense Evergreen Needleleaf Forests
Level L0
Level L1
Level L2
Level L3
Level L4
Level L5
Levels "L1" AND "L2" AND "L3" AND "L4" AND "L5"
Levels "L2" AND "L3" AND "L4" AND "L5"
Levels "L3" AND "L4" AND "L5"

Figure 4.2: Tree representation of the six-level (L0 to L5) hierarchical structure of the Land-Use and Land-Cover (LULC) classes contained in the Sentinel2GlobalLULC dataset. Outter circular leafs represent the final or most detailed 29 LULC classes of level L5. The followed path to define each class is represented through inner ellipses that contain the names of intermediate classes at different levels between the division of the Earth's surface (square) into LU and LC (level L0) and the final class circle (level L5). All LULC classes belong to three levels at least, except the 12 forest classes that belong to L5 only.

4.3.1.3 Combining products across time and space

For each one of the 29 LULC classes, we combined in space and time the global LULC information among the 15 GEE LULC products. For each product and LULC type, we first set one or more criteria to create a global mask at the native resolution of the product in which each pixel was classified as 0 or 1 depending on whether it met the criteria for belonging to that LULC type or not, respectively (see first stage in Table 4.3). Then (see second stage in Table 4.4), for each LULC type, we calculated the average of all the masks obtained from each product to create a final global probability map at the finest resolution from all products with values ranging between 0 and 1. Value 1 meant that all products

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15
C1	16	15	NA	7	11	60	$TCC < 10$	200	0	2	$(TC < 10) \cap (G = 0) \cap (L = 0) \cap (D \neq 2)$	$TH < 1$	$1 \cup 0$	0	Not($\geq$ 1)
C2	16	15	NA	7	11	NA	$TCC < 10$	$200 \cup 150$	0	2	$(TC < 10) \cap (G = 0) \cap (L = 0) \cap (D \neq 2)$	$TH < 1$	$1 \cup 0$	0	Not($\geq$ 1)
C3	10	10	1	6	6	30	$TCC < 10$	140	NA	2	$(TC < 10) \cap (G = 0) \cap (L = 0) \cap (D \neq 2)$	$TH < 2$	$1 \cup 0$	0	Not($\geq$ 1)
C4	7	7	2	NA	5	$20 \cap (10 < SCF < 50)$	$TCC < 10$	150	0	2	$(TC < 10) \cap (G = 0) \cap (L = 0) \cap (D \neq 2)$	$TH < 2$	$1 \cup 0$	0	Not($\geq$ 1)
C5	6	6	2	NA	5	$20 \cap (SCF > 50)$	$TCC < 10$	130	0	2	$(TC < 10) \cap (G = 0) \cap (L = 0) \cap (D \neq 2)$	$TH < 2$	$1 \cup 0$	0	Not($\geq$ 1)
C6	NA	NA	NA	4	4	$4 + (15 < TCF < 30)$	$15 < TCC < 30$	60	NA	1	$(15 < TC < 30) \cap (G = 0) \cap (L = 0) \cap (D \neq 2)$	$TH > 2$	$1 \cup 0$	0	Not($\geq$ 1)
C7	NA	NA	NA	4	4	$4 + (40 < TCF < 60)$	$40 < TCC < 60$	50	NA	1	$(40 < TC < 60) \cap (G = 0) \cap (L = 0) \cap (D \neq 2)$	$TH > 2$	$1 \cup 0$	0	Not($\geq$ 1)
C8	4	4	6	4	4	$4 + (TCF > 60)$	$TCC > 60$	50	NA	1	$(TC > 60) \cap (G = 0) \cap (L = 0) \cap (D \neq 2)$	$TH > 2$	$1 \cup 0$	0	Not($\geq$ 1)
C9	NA	NA	NA	3	3	$3 + (15 < TCF < 30)$	$15 < TCC < 30$	NA	NA	1	$(15 < TC < 30) \cap (G = 0) \cap (L = 0) \cap (D \neq 2)$	$TH > 2$	$1 \cup 0$	0	Not($\geq$ 1)
C10	NA	NA	NA	3	3	$3 + (40 < TCF < 60)$	$40 < TCC < 60$	NA	NA	1	$(40 < TC < 60) \cap (G = 0) \cap (L = 0) \cap (D \neq 2)$	$TH > 2$	$1 \cup 0$	0	Not($\geq$ 1)
C11	3	3	8	3	3	$3 + (TCF > 60)$	$TCC > 60$	NA	NA	1	$(TC > 60) \cap (G = 0) \cap (L = 0) \cap (D \neq 2)$	$TH > 2$	$1 \cup 0$	0	Not($\geq$ 1)
C12	NA	NA	NA	2	2	$2 + (15 < TCF < 30)$	$15 < TCC < 30$	40	NA	1	$(15 < TC < 30) \cap (G = 0) \cap (L = 0) \cap (D \neq 2)$	$TH > 2$	$1 \cup 0$	0	Not($\geq$ 1)
C13	NA	NA	NA	2	2	$2 + (40 < TCF < 60)$	$40 < TCC < 60$	40	NA	1	$(40 < TC < 60) \cap (G = 0) \cap (L = 0) \cap (D \neq 2)$	$TH > 2$	$1 \cup 0$	0	Not($\geq$ 1)
C14	2	2	5	2	2	$2 + (TCF > 60)$	$TCC > 60$	40	NA	1	$(TC > 60) \cap (G = 0) \cap (L = 0) \cap (D \neq 2)$	$TH > 2$	$1 \cup 0$	0	Not($\geq$ 1)
C15	9	9	NA	1	1	$1 + (15 < TCF < 30)$	$15 < TCC < 30$	90	NA	1	$(15 < TC < 30) \cap (G = 0) \cap (L = 0) \cap (D \neq 2)$	$TH > 2$	$1 \cup 0$	0	$Not(\geq 1)$
C16	8	8	4	1	1	$1 + (40 < TCF < 60)$	$40 < TCC < 60$	70	NA	1	$(40 < TC < 60) \cap (G = 0) \cap (L = 0) \cap (D \neq 2)$	$TH > 2$	$1 \cup 0$	0	Not($\geq$ 1)
C17	1	1	7	1	1	$1 + (TCF > 60)$	$TCC > 60$	70	NA	1	$(TC > 60) \cap (G = 0) \cap (L = 0) \cap (D \neq 2)$	$TH > 2$	$1 \cup 0$	0	Not($\geq$ 1)
C18	11	11	NA	NA	NA	90	$TCC > 10$	170	NA	NA	$(TC > 10) \cap (G = 0) \cap (L = 0) \cup (D = 2)$	$TH > 2$	$2 \cup 3$	1	Not($\geq$ 1)
C19	11	11	NA	NA	NA	90	$TCC > 10$	$a.160 \cup 180$ $b.Not(170)$	NA	NA	$(TC > 10) \cap (G = 0) \cap (L = 0) \cup (D = 2)$	$TH > 2$	$2 \cup 3$	1	Not($\geq$ 1)
C20	11	11	NA	NA	NA	90	$TCC < 10$	$160 \cup 170$ $\cup 180$	NA	NA	$(TC < 10) \cap (G = 0) \cap (L = 0) \cup (D = 2)$	$TH < 2$	$2 \cup 3$	1	Not($\geq$ 1)
C21	17	0	0	0	0	200	NA	210	NA	3	NA	NA	3	1	Not($\geq$ 1)
C22	17	0	0	0	0	80	NA	210	NA	3	NA	NA	3	1	Not($\geq$ 1)
C23	15	NA	NA	NA	10	70	NA	220	NA	NA	NA	NA	$1 \cup 0$	0	Not($\geq$ 1)
C24	12	12	$3 \cup 1$	$5 \cup 6$	$7 \cup 8$	40	NA	$11 \cup 14$	$1 \cup 2 \cup 3$ $\cup 4 \cup 5$	NA	NA	NA	$2 \cup 3$	$0 \cup 4 \cup$ $\cup 8 \cup 10$	Not($\geq$ 1)
C25	12	12	1	6	7	40	NA	11	$1 \cup 2$	NA	NA	NA	$1 \cup 0$	0	Not($\geq$ 1)
C26	12	12	1	6	7	40	NA	14	$3 \cup 4 \cup 5$	NA	NA	NA	$1 \cup 0$	0	Not($\geq$ 1)
C27	12	12	3	5	8	40	NA	11	$1 \cup 2$	NA	NA	NA	$1 \cup 0$	0	Not($\geq$ 1)
C28	12	12	3	5	8	40	NA	14	$3 \cup 4 \cup 5$	NA	NA	NA	$1 \cup 0$	0	Not($\geq$ 1)
C29	13	13	10	8	9	50	NA	190	NA	NA	NA	NA	$1 \cup 0$	0	NU

Table 4.3: First stage of the rule set criteria used to find consensus across the 15 Land-Use and Land-Cover (LULC) products available in GEE for each of the 29 LULC classes contained in the Sentinel2GlobalLULC dataset. P1 to P15: product 1 to 15. C1 to C29: class 1 to class 29. For each product, one or multiple criteria were established to create a global probability map (pixel values 0 or 1) for a given LULC class. A total number of 15x29 = 435 of global probability maps were calculated. The numbers in each column (i.e., from 0 to 220) correspond to the pixel values from each product band. NU: Not Used, NA: Not Available, TC: Tree Cover, G: Tree Gain, L: Tree Loss, D: Datamask, TH: Tree Hight, TCC: Tree Canopy Cover, TCF: Tree-Cover Fraction, and SCF: Shrub-Cover Fraction. ∩:"AND", ∪:"OR" , +:"ADD".

agreed to assign that pixel to a particular class and value 0 meant that none of the products assigned it to that particular class (Figure 4.3). These 0-to-1 values are interpreted as the spatio-temporal purity level of each pixel to belong to a particular LULC class.

As an example of the first stage (see details in Table 4.3), to specify if a given pixel belongs to a dense, evergreen or needleleaf forest, we evaluated its tree cover level using "$\leq$" and "$\geq$", while for bands containing the leaf type information, we used the equal operator "$=$". For the spatio-temporal combination of multiple criteria we have used the following operators: "AND","OR" and "ADD". For example, we combined the tree cover percentage criteria with the leaf type criteria using "AND" in order to select forest pixels that meets both conditions. To combine many years instances of the same product we used "ADD", except for product P13 where we used "AND" to select permanent water areas. Whenever we used the "ADD" operator, we normalized pixel values afterwards to bring it back to a probability interval between 0 and 1 using the division by the total number of combined years or criteria.

In the second stage (see details in Table 4.4), we combined for each LULC class, the 15 global probability maps resulted from the previous stage to create a final global probability map. This combination was carried out using various operators such as "ADD", "MULTIPLY" and "OR", depending on the LULC type. When "ADD" was used, the final pixel values were normalized by dividing the final addition value of each pixel by the total number of added products. The "MULTIPLY" operator was mostly used at the end, to remove urban areas from non-urban LULC classes, or to remove water from non-water systems. The multiplication operator was also adopted to make sure that a certain criteria was respected in the final probability map. For instance, for the swamp class, we multiplied all pixels in the final stage by a water mask where saline water areas have a value of 0 in order to eliminate mangroves from swamp pixels and vice versa. Finally, we used "OR" operator between different water related products in order to take advantage of the fact that each one complements the other in terms of spatial coverage and accuracy.

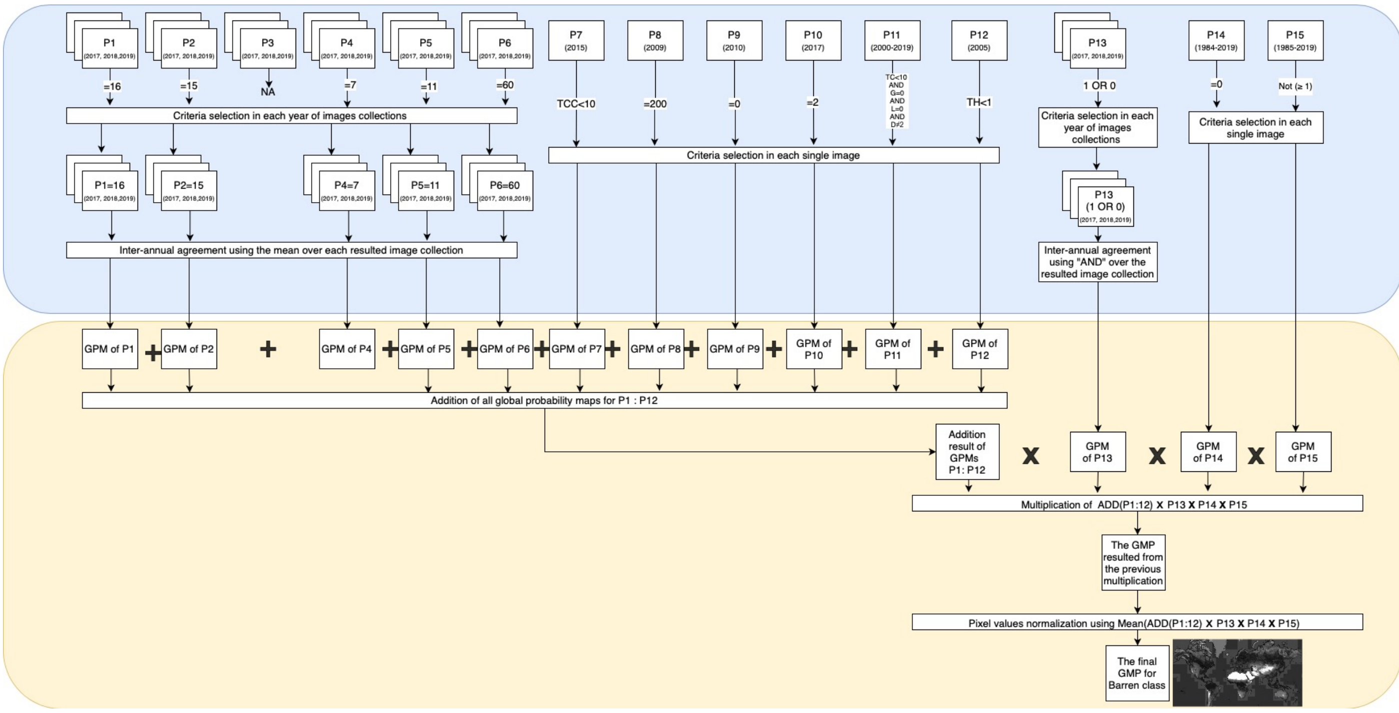

P1 (2017, 2018,2019)
P2 (2017, 2018,2019)
P3 (2017, 2018,2019)
P4 (2017, 2018,2019)
P5 (2017, 2018,2019)
P6 (2017, 2018,2019)
P7 (2015)
P8 (2009)
P9 (2010)
P10 (2017)
P11 (2000-2019)
P12 (2005)
P13 (2017, 2018,2019)
P14 (1984-2019)
P15 (1985-2019)
=16
=15
NA
=7
=11
=60
TCC<10
=200
=0
=2
TC<10 AND G=0 AND L=0 AND D≠2
TH<1
1 OR 0
=0
Not (≥ 1)
Criteria selection in each year of images collections
Criteria selection in each single image
Criteria selection in each year of images collections
Criteria selection in each single image
P1=16 (2017, 2018,2019)
P2=15 (2017, 2018,2019)
P4=7 (2017, 2018,2019)
P5=11 (2017, 2018,2019)
P6=60 (2017, 2018,2019)
P13 (1 OR 0) (2017, 2018,2019)
Inter-annual agreement using the mean over each resulted image collection
Inter-annual agreement using "AND" over the resulted image collection
GPM of P1
+
GPM of P2
+
GPM of P4
+
GPM of P5
+
GPM of P6
+
GPM of P7
+
GPM of P8
+
GPM of P9
+
GPM of P10
+
GPM of P11
+
GPM of P12
Addition of all global probability maps for P1 : P12
Addition result of GPMs P1: P12
X
GPM of P13
X
GPM of P14
X
GPM of P15
Multiplication of ADD(P1:12) X P13 X P14 X P15
The GMP resulted from the previous multiplication
Pixel values normalization using Mean(ADD(P1:12) X P13 X P14 X P15)
The final GMP for Barren class

Figure 4.3: Example of the process of building the final global probability map for one of the 29 Land-Use and Land-Cover (LULC) classes (e.g. C1: "Barren") by means of spatio-temporal agreement of the 15 LULC products available in GEE. The final map is normalized to values between 0 (white, i.e., areas with no presence of C1 in any product) and 1 (black spots, i.e., areas containing or compatible with the presence of C1 in all 15 products), whereas the shades of grey corresponds to the values in between (i.e., areas that did not contain or were not compatible with the presence of C1 in some of the products). This process is divided into two stages: the first stage (the blue part, see details in Table 4.3) and the second stage (the yellow part, see details in Table 4.4). LULC products available for several years are represented with superposed rectangles, while single year products are represented with single rectangles. GMP: global probability map, NA: Not Available.

4.3.1.4 Re-projection and Selection of purity threshold

After the consensus assessment, the 29 final probability maps maintained the spatial resolution of the last aggregated LULC product, i.e., the water product at 30m/pixel. Since our objective was finding pure tiles of 224 × 224 10-m pixels (i.e. Sentinel-2 pixels) for each LULC class, we reprojected the 30 m/pixel probability maps to 2240 m/pixel by using the spatial mean reducer in GEE.

For each one of the reprojected maps, we defined a pixel value threshold to decide whether a given 2240 × 2240 m tile was representative of each LULC class or not. If the number of available pure tiles (i.e., pixel value = 1) was too small for one class, we decreased the threshold for purity level for that class until getting a large enough number of tiles (the purity level is always provided as metadata for each tile). On the other hand, when the number of pure tiles for a LULC class was too large, (i.e., greater than 14000), we applied a stratified selection to download a maximum of 14000 images. This selection was carried out through an automatic maximum geographic distance algorithm to guarantee

Class ID	LULC class	Spatial Combination
C1	Barren lands	Norm(Add(P1:P12)*P13*P14*P15)
C2	Moss and Lichen lands	Norm(Add(P1:P12)*P13*P14*P15)
C3	Grasslands	Norm(Add(P1:P12)*P13*P14*P15)
C4	Open Shrublands	Norm(Add(P1:P12)*P13*P14*P15)
C5	Close Shrublands	Norm(Add(P1:P12)*P13*P14*P15)
C6	Open Deciduous Broadleaf Forests	Norm(Add(P1:P12)*P13*P14*P15)
C7	Close Deciduous Broadleaf Forests	Norm(Add(P1:P12)*P13*P14*P15)
C8	Dense Deciduous Broadleaf Forests	Norm(Add(P1:P12)*P13*P14*P15)
C9	Open Deciduous Needleleaf Forests	Norm(Add(P1:P12)*P13*P14*P15)
C10	Close Deciduous Needleleaf Forests	Norm(Add(P1:P12)*P13*P14*P15)
C11	Dense Deciduous Needleleaf Forests	Norm(Add(P1:P12)*P13*P14*P15)
C12	Open Evergreen Broadleaf Forests	Norm(Add(P1:P12)*P13*P14*P15)
C13	Close Evergreen Broadleaf Forests	Norm(Add(P1:P12)*P13*P14*P15)
C14	Dense Evergreen Broadleaf Forests	Norm(Add(P1:P12)*P13*P14*P15)
C15	Open Evergreen Needleleaf Forests	Norm(Add(P1:P12)*P13*P14*P15)
C16	Close Evergreen Needleleaf Forests	Norm(Add(P1:P12)*P13*P14*P15)
C17	Dense Evergreen Needleleaf Forests	Norm(Add(P1:P12)*P13*P14*P15)
C18	Mangrove Wetlands	Norm(Add(P1:P7,P9:P14)*P8*P15)
C19	Swamp Wetlands	Norm(Add(P1:P7,a.P8,P9:P14)*b.P8*P15)
C20	Marshland Wetlands	Norm(Add(P1:P6,P8:P10,P13,P14)*(P11 OR P12 OR P7)*P15)
C21	Marine Water Bodies	Norm(Add(P1:P12)*P13*P14*P15)
C22	Continental Water Bodies	Norm(Add(P1:P12)*P13*P14*P15)
C23	Permanent Snow	Norm(Add(P1:P12)*P13*P14*P15)
C24	Croplands Flooded with seasonal water	Norm(Add(P1:P12)*(P13 OR P14)*P15)
C25	Cereal Irrigated Cropland	Norm(Add(P1:P12)*P13*P14*P15)
C26	Cereal Rainfed Cropland	Norm(Add(P1:P12)*P13*P14*P15)
C27	Irrigated Broadleaf Cropland	Norm(Add(P1:P12)*P13*P14*P15)
C28	Rainfed Broadleaf Cropland	Norm(Add(P1:P12)*P13*P14*P15)
C29	Urban and built-up areas	Norm(Add(P1:P12)*P13*P14*P15)

Table 4.4: Second stage of the rule set criteria used to find consensus across the 15 Land-Use and Land-Cover (LULC) products available in GEE for each of the 29 LULC classes contained in the Sentinel2GlobalLULC dataset. P1 to P15: product 1 to 15. C1 to C29: class 1 to class 29. For each LULC class, the 15 global probability maps (with pixel values 0 or 1) obtained in the first stage from products P1 to P15 were spatially combined to build 29 final global probability maps (with pixel values 0 to 1), one for each LULC class (C1 to C29). "Add":ADD, "*":MULTIPLY, "Norm": the normalization using division by number of used products

LCLU Class	Consensus probability values (%)						Number of selected images	Stratified selection
	0.75 (75%)	0.80 (80%)	0.85 (85%)	0.90 (90%)	0.95 (95%)	1.00 (100%)		
Urban	63953	-	34102	21814	12590	192	12590	no
Barren	4330418	-	4055836	3876467	3545756	2668009	**14000 (2668009)**	yes
Moss and Lichen	59120	-	18438	4669	1158	0	4669	no
Close Shrublands	41407	12502	1872	226	16	0	12502	no
Open Shrublands	2461415	-	1209375	644272	101288	805	**14000 (101288)**	yes
Marshland	4205	-	675	143	15	0	4205	no
Swamp	489	-	4	0	0	0	489	no
Mangrove	425	-	63	3	0	0	425	no
Grassland	4022949	-	1894337	943177	128263	8895	8895	no
Rainfed Broadleaf Cropland	427314	-	209143	99337	32123	416	416	no
Irrigated Broadleaf Cropland	224867	-	92488	53064	30691	354	354	no
Cereal Rainfed Cropland	1185497	-	604459	284914	91147	1022	1022	no
Cereal Irrigated Cropland	517789	-	167994	52959	23555	842	842	no
Cropland Seasonal Water	6048	-	3192	2008	995	15	2008	no
Dense Evergreen Needleleaf Forest	474138	-	178293	66151	13995	0	13995	no
Close Evergreen Needleleaf Forest	43040	3875	69	0	0	0	3875	no
Open Evergreen Needleleaf Forest	17462	3939	331	0	0	0	3939	no
Dense Evergreen Broadleaf Forest	2131269	-	1829897	1594657	1232914	144026	**14000 (144026)**	yes
Close Evergreen Broadleaf Forest	12512	1270	77	1	0	0	1270	no
Open Evergreen Broadleaf Forest	574	42	0	0	0	0	574	no
Dense Deciduous Needleleaf Forest	60866	-	12954	2888	148	0	2888	no
Close Deciduous Needleleaf Forest	42166	6383	35	0	0	0	6383	no
Open Deciduous Needleleaf Forest	10439	23	0	0	0	0	10439	no
Dense Deciduous Broadleaf Forest	399264	-	176176	97182	31284	1	**14000 (31284)**	yes
Close Deciduous Broadleaf Forest	71127	-	1353	23	1	0	1353	no
Open Deciduous Broadleaf Forest	25342	4439	466	2	0	0	4439	no
Permanent Snow	1065127	-	1033466	1013490	984014	877232	**14000 (877232)**	yes
Continental Water Bodies	3543953	-	3199652	343779	318483	265214	**14000 (265214)**	yes
Marine Water Bodies	3606955	-	3357810	2903459	2822544	2577444	**14000 (2577444)**	yes

Table 4.5: Summary of the varying number of found and eventually selected Sentinel-2 image tiles of 224 × 224 pixels depending on the different consensus level reached across the 15 Land-Use and Land-Cover (LULC) products available in GEE for each of the 29 LULC classes contained in the Sentinel2GlobalLULC dataset. LULC classes that due to the too large number of samples had to undergo a stratified selection by maximizing geographical distance among samples are highlighted in bold.

that selected images were as geographically far from each other as possible. In Table 4.5, we present the number of tiles we found and downloaded for each LULC class using thresholds ranging from 0.75 to 1. We illustrated the reprojection and selection processes in Figure 4.4.

4.3.2 Data Extraction

Sentinel2GlobalLULC provides the user with two types of data: CSV files and Sentinel-2 RGB images. In the following subsections, we first present the additional gHM index attached to these both data types, then the adopted methods to generate each one of them.

4.3.2.1 Global human modification (gHM) values extraction

As an additional metadata related to the level of human influence in each image, we calculated for each tile the spatial mean of the global human modification (gHM) index for terrestrial lands [90], where 0 means no human modification and 1 means complete transformation. Since the original gHM product was mapped at 1 × 1 km resolution, we reprojected it to 2240 × 2240 m using the same procedure than explained for the LULC consensus masks.

4.3.2.2 Comma-separated values (CSV) files generation

Once we identified tiles to be selected for each LULC class, we have grouped their center coordinates into a CSV file each. Tiles were organized giving their probability values in an descendant order. Each row in the CSV file corresponds to a selected tile in that class. In fact, these CSV files contains the geographical center point coordinates, the pixel purity value, the name of the attributed LULC class in addition to the extracted gHM value for that tile. Then, we used the geographical coordinates of each tile to identify its exact administrative address geolocation. To implement this reverse geo-referencing operation, we used a free request-unlimited python module called reverse_geocoder. This method has allowed us to identify the country code, the administrative department at two levels and the locality of each tile in the CSV files. This way, we integrated in all LULC classes CSV files these reverse geo-referencing information as new columns.

For LULC class that has more than 14000 pure tiles, we have released the coordinates before and after the stratified selection in case the user was interested in all tiles and not only the exported ones. These coordinates could allow the end user to download new images if needed.

4.3.2.3 Sentinel-2 RGB images exportation

After extracting all these pieces of information and grouping them into CSV files, we went back to the geographic center coordinates of each tile and used them to extract the corresponding 224 × 224 pixel Sentinel-2 RGB tiles using GEE. Each exported image was identical to the 2240 × 2240 m area covered by its Sentinel-2 tile.

We chose ”Sentinel-2 MSI (Multi-Spectral Instrument) product” since it is free and publicly available in GEE at the fine resolution of 10 × 10 m. We chose ”Level-1C” since it provides the longest data availability of Sentinel-2 images. To build RGB images, we extracted the three bands B4, B3 and B2 that correspond to Red, Green and Blue channels, respectively.

To minimize the effect of atmospheric effects on the RGB images, such as clouds, aerosols, smoke, etc., every image was built from the 25th-percentile aggregation of its corresponding image collection gathered by Sentinel-2 satellites between June 2015 and October 2020. In addition, we previously discarded all pixels where the maximum cloud probability exceeded 20% according to the metadata provided in the Sentinel-2 collection.

Usually, Sentinel-2 MSI product includes true colour images in JPEG2000 format, except for the ”Level-1C” collection used here. The three original bands (B4, B3, and B2) required a saturation stretching of their reflectance values into 0-255 RGB digital values. Thus, we stretched the saturation reflectance of 3558 into 255 to obtaine true RGB channels with digital values between 0 and 255. The choice of these mapping numbers was taken from the Sentinel-2 true colour image recommendations section of Sentinel user guidelines. Finally, after exporting the selected tiles for each LULC class as ”.tif” images, we converted them into ”.jpeg” format using a lossless conversion algorithm.

4.3.3 Technical implementation

To implement all our methodology steps, we first created a javascript in GEE for each LULC class. Each script is a multi-task javascript where we implemented a switch command to control which task we want to execute. In each one of these scripts, we selected from GEE LULC datasets repository the 15 LULC products used to build the consensus of that LULC class. Each script was responsible of elaborating the spatio-temporal combination of the selected products and generating the final consensus map for that LULC class as described in the subsection Combining products across time and space. Then, it exports the final global probability map as an asset into GEE server storage to make its reprojection faster. In the same script, once the consensus map exportation was done, we imported it from the GEE assets storage and reprojected it to 2240 × 2240 m resolution; then, we exported the new reprojected map into GEE assets storage again to make its analysis and processing faster. Afterwards, we imported the reprojected map into the same script and apply different processing tasks. During this processing phase, many purity threshold values were evaluated. Then, we elaborated in this same script the pure tiles identification and their center coordinates exportation into a CSV file. A distinct GEE script was developed to import, reproject and export the global gHM map. The resulted gHM map was saved as an asset, then imported and used in each one of the 29 LULC multi-task scripts.

A python script was developed separately to read the exported CSV files for each LULC class and apply the reverse geo-referencing on their pure tiles coordinates then add the found geolocalization data (country code, locality...etc) to the original CSV files as new columns. Then, another python script was implemented to read the new resulted CSV files with all their added columns (reverse geo-referencing data, gHM data) and use the center coordinates of each pure tile in that class to export its corresponding Sentinel-2 satellite tif image within GEE through the python API. Finally, after downloading all the exported tif images from our Google drive, we created another python script to convert the exported tif images into JPEG format.

4.4 Data Records

Sentinel2GlobalLULC dataset is stored in the following Zenodo repository(DOI:10.5281/zenodo.5055632). This dataset consists of three zip compressed folders:

- Sentinel-2 GeoTiff images folder: This folder contains the exported Sentinel-2 RGB images for each LULC class grouped into sub-folders named according to each LULC class. Each image has a filename with the following structure: ”LULC class ID_LULC class short name_Pixel probability value_Image ID_GHM value_Latitude _Longitude_Country code_Administrative department level1_Administrative department level2_Locality”. Pixel probability value can be interpreted as the spatial purity of the image to represent that LULC class and was calculated as the spatial mean of all the pixels of the final probability maps contained in each image tile, reprojected and expressed as a percentage. Short names for all classes were derived from the original ones in a way to have exactly 13 characters each, and IDs for different classes were assigned randomly. This information for each class is explained in Table 4.6.

- Sentinel-2 JPEG images folder: This folder contains the same images as in the GeoTiff folder, but converted into ”.jpeg” format while preserving the same nomenclature and organization. In Figure 4.5, we illustrate a sample of each one of the 29 classes images in JPEG format.

- CSV files folder: For user convenience, the metadata of every image tile (i.e., the same information already contained in the image filenames) is also provided in CSV format. Image tiles in the CSV files are organized from the highest to the lowest consensus probability value. These CSV files have 11 columns: ID of LULC Class, Short name of LULC Class, ID Image, Pixel Probability Value as percentage, GHM Value, Center Latitude, Center Longitude, Country Code, Administative Departement Level 1, Administative Departement Level 2, Locality.

  For too large LULC classes (i.e., with more than 14000 potential samples) that had to undergo a stratified selection, we provide the user with 2 CSV files: one containing all pure tiles coordinates without geo-referencing columns, and another file just containing the 14000 exported tiles coordinates with their geo-referencing information.

LCLU Class	**Short name**	**Class ID**
Urban	UrbanBlUpArea	29
Barren	BarrenLands__	1
Moss and Lichen	MossAndLichen	2
Close Shrublands	SrublandClose	5
Open Shrublands	ShrublandOpen	4
Marshland	WetlandMarshl	20
Swamp	WetlandSwamps	19
Mangrove	WetlandMangro	18
Grassland	Grasslands___	3
Rainfed Broadleaf Cropland	CropBroadRain	28
Irrigated Broadleaf Cropland	CropBroadIrri	27
Cereal Rainfed Cropland	CropCereaRain	26
Cereal Irrigated Cropland	CropCereaIrri	25
Cropland Seasonal Water	CropSeasWater	24
Dense Evergreen Needleleaf Forest	ForestsDeEvNe	17
Close Evergreen Needleleaf Forest	ForestsClEvNe	16
Open Evergreen Needleleaf Forest	ForestsOpEvNe	15
Dense Evergreen Broadleaf Forest	ForestsDeEvBr	14
Close Evergreen Broadleaf Forest	ForestsClEvBr	13
Open Evergreen Broadleaf Forest	ForestsOpEvBr	12
Dense Deciduous Needleleaf Forest	ForestsDeDeNe	11
Close Deciduous Needleleaf Forest	ForestsClDeNe	10
Open Deciduous Needleleaf Forest	ForestsOpDeNe	9
Dense Deciduous Broadleaf Forest	ForestsDeDeBr	8
Close Deciduous Broadleaf Forest	ForestsClDeBr	7
Open Deciduous Broadleaf Forest	ForestsOpDeBr	6
Permanent Snow	PermanentSnow	23
Continental Water Bodies	WaterBodyCont	22
Marine Water Bodies	WaterBodyMari	21

Table 4.6: Dictionary to map each Land-Use and Land-Cover (LULC) class to its corresponding short name and ID in the Sentinel2GlobalLULC dataset

L0	F1	L1	F1	L2	F1	L3	F1	L4	F1	L5	F1
Land Cover	0.99	Terrestrial Lands	1.00	BarrenLands	0.97	BarrenLands	0.97	BarrenLands	0.97	BarrenLands	0.97
				MossAndLichen	NA	MossAndLichen	NA	MossAndLichen	NA	MossAndLichen	NA
				Grasslands	0.75	Grasslands	0.75	Grasslands	0.75	Grasslands	0.75
				Shrubland	0.89	ShrublandOpen	0.76	ShrublandOpen	0.76	ShrublandOpen	0.76
						SrublandClose	0.97	SrublandClose	0.97	SrublandClose	0.97
				Forests	1.00	ForestsDe	1.00	ForestsDeBr	1.00	ForestsOpDeBr	0.82
										ForestsCIDeBr	0.89
										ForestsDeDeBr	0.96
								ForestsDeNe	1.00	ForestsOpDeNe	0.92
										ForestsCIDeNe	0.88
										ForestsDeDeNe	0.95
						ForestsEv	0.99	ForestsEvBr	0.99	ForestsOpEvBr	0.70
										ForestsCIEvBr	0.72
										ForestsDeEvBr	0.91
								ForestsEvNe	1.00	ForestsOpEvNe	0.82
										ForestsCIEvNe	0.88
										ForestsDeEvNe	0.99
				PermanentSnow	1.00	PermanentSnow	1.00	PermanentSnow	1.00	PermanentSnow	1.00
		Aquatic Lands	0.98	Wetland	0.96	WetlandMangro	0.96	WetlandMangro	0.96	WetlandMangro	0.96
						WetlandSwamps	0.99	WetlandSwamps	0.99	WetlandSwamps	0.99
						WetlandMarshl	0.94	WetlandMarshl	0.94	WetlandMarshl	0.94
				WaterBody	0.99	WaterBodyMari	0.95	WaterBodyMari	0.95	WaterBodyMari	0.95
						WaterBodyCont	0.93	WaterBodyCont	0.93	WaterBodyCont	0.93
Land Use	0.98	Croplands	0.98	CropSeasWater	0.93	CropSeasWater	0.93	CropSeasWater	0.93	CropSeasWater	0.93
				CropCerea	0.99	CropCereaIrri	1.00	CropCereaIrri	1.00	CropCereaIrri	1.00
						CropCereaRain	0.98	CropCereaRain	0.98	CropCereaRain	0.98
				CropBroad	0.99	CropBroadIrri	1.00	CropBroadIrri	1.00	CropBroadIrri	1.00
						CropBroadRain	0.99	CropBroadRain	0.99	CropBroadRain	0.99
		UrbanBlUpArea	0.99	UrbanBlUpArea	0.99	UrbanBlUpArea	0.99	UrbanBlUpArea	0.99	UrbanBlUpArea	0.99
Mean	0.99		0.98		0.95		0.95		0.95		0.91

Table 4.7: Results of the validation procedure of the representativeness of the images contained in the Sentinel2GlobalLULC dataset for each Land-Use and Land-Cover (LULC) class at different levels of the hierarchical legend (from L0 to L5). Accuracy is expressed as the mean F1 score (i.e., a balance between precision and recall) for each LULC class at each level, rounded to two decimal values.

4.5 Technical Validation

To assess the quality of the Sentinel2GlobalLULC dataset in terms of its representativity of each LULC class and of image quality, two of the coauthors visually inspected very high resolution imagery (Google Earth and Bing Maps) of a random sample of each class. The validation process was established in three stages:

- First, for each LULC class, we selected 100 samples to visually verify their LULC annotation. To maximize the global representativity of the validated samples, the selection of these 100 samples was carried out by maximizing the geographical distance among all samples using an add-hoc script in R. In Figure4.6, we present the distribution map of the 100 samples selected for each LULC class.

- Second, each one of the selected samples was visually inspected in Google Earth and Bing Maps by two of the co-authors (E.G. and D.A-S.) to independently assign it to one of the 29 LULC classes. These two experts assigned each sample to a LULC class when it occupied more than 70% of the image tile.

- Third, the confusion matrix for this validation was calculated at six different levels of our LULC classification hierarchy (from L0 to L5 as presented in Figure4.2). In Table4.7, we summarized the obtained F1 scores at each level.

The obtained mean F1 scores ranged from 0.99 at level L0 to 0.91 at level L5 (Table4.7). Such decrease in accuracy as the number of classes increased from level L0 to level L5 was mainly due to the hard distinction between forest types at L5 and the complexity of visual features in Grasslands and Shrublands classes from level L2.

4.6 Usage Notes and code availability

To make the Sentinel2GlobalLULC dataset easier to use, reproduce, and exploit and to promote its usage with DL models, we have provided users with a python code to load all RGB images and train several Convolutional Neural Networks (CNNs) models on them using different learning hyper-parameters. Knowing that most CNN frameworks admit only jpeg or png images formats, we provided a python script to convert ".tif" into ".jpeg" format with a full control on the conversion quality and the choice of images to convert. Moreover, as for some LULC classes we limited the number of exported images to 14000, we have provided a python script that can help the user to export more Sentinel-2 images of these classes if needed, using the coordinates stored in the CSV files.

In addition, to provide a global insight about the consistency and accuracy of the global distribution of these 29 LULC classes, we also publicly shared the final reprojected global consensus maps for each class as GEE assets. To help the user to visualize the global distribution of each LULC class, we have provided a GEE script with the assets links to choose, import, manipulate, and visualize any LULC class map. Image exportation is also possible through python API for GEE and we gave the user a complete control on the number of tiles to export, the time interval to select for image collections, the cloud removal parameters, the true RGB colors calibration, and the Google drive account where to store the exported images. The user should be aware that GEE imposes a limited request number with a maximum of 3000 exportation tasks to run simultaneously on the same Google account.

All used scripts to implement our dataset and links to the GEE stored assets are available in the following Github repository (DOI:10.5281/zenodo.5638409) with guidelines stored in a README file explaining all instructions about their execution.

4.7 Conclusion

In this chapter, we have created a new free, smart and global LULC mapping dataset called Sentinel2GlobalLULC. This dataset will help researchers in the creation of accurate global LULC products by using DL models and techniques. We expect this dataset to improve our understanding and modelling of natural and human systems around the world. On the other side, in the context of this thesis, all the used methodology to create Sentinel2GlobalLULC as a high quality dataset constitutes in itself the preprocessing that will prepare the data to be used with DL models. In fact, after evaluating DL performance on an already available dataset in another complex problem which is breast cancer based CAD systems; this dataset that we have created will allow us to establish a different analysis of our hypothesis but under a new scenario. In this scenario we have our own dataset that we made deep-learning-ready as much as possible with all the preprocessing methods and preparation techniques used during its creation. Another important point that will be addressed with this dataset is to verify either the data structure and the classification task that it was originally built for, affect DL models performance or not. In the next chapter, we will analyze the performance of multiple DL models on Sentinel2GlobalLULC dataset for the complex problem of LULC classification at the global scale.

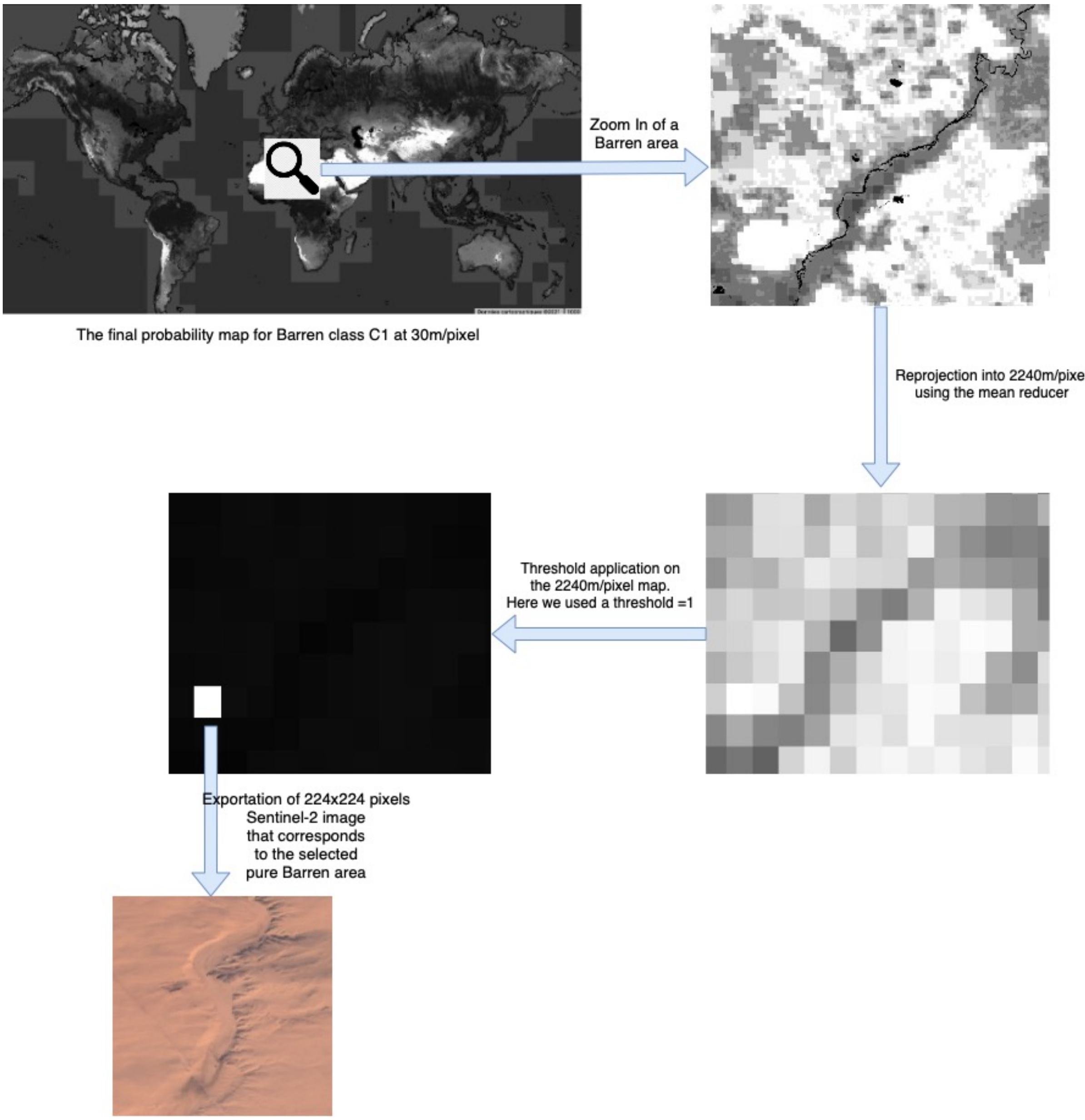
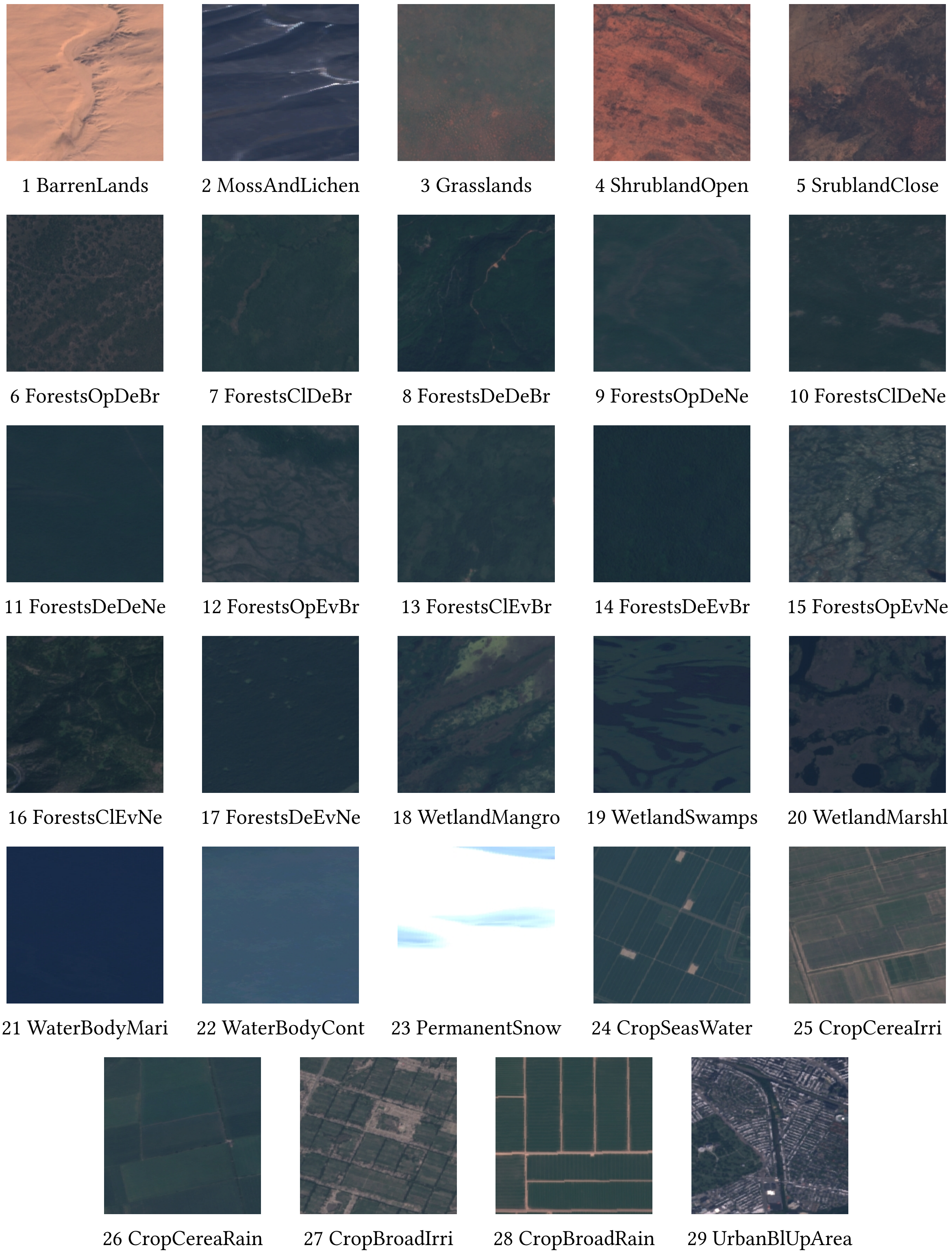

The final probability map for Barren class C1 at 30m/pixel
Zoom In of a Barren area
Reprojection into 2240m/pixel using the mean reducer
Threshold application on the 2240m/pixel map. Here we used a threshold =1
Exportation of 224x224 pixels Sentinel-2 image that corresponds to the selected pure Barren area

Figure 4.4: Example of the workflow to obtain a Sentinel-2 image tile of 2240 × 2240 m for one of the 29 Land-Use and Land-Cover (LULC) classes (e.g. C1: ”Barren”). The process starts with the reprojected final global probability map obtained from stage two (Table 4.4) and ends with its exportation to the repository of a Sentinel-2 image tile of 224 × 224 pixels. The white rectangle is the only one having a probability value of 1 (Recall that the purity threshold used for Barren was 1, i.e., 100%). The black pixels has a null probability value, while the probability values between 0 and 1 are represented in gray scale levels.



1 BarrenLands 2 MossAndLichen 3 Grasslands 4 ShrublandOpen 5 SrublandClose

6 ForestsOpDeBr 7 ForestsClDeBr 8 ForestsDeDeBr 9 ForestsOpDeNe 10 ForestsClDeNe

11 ForestsDeDeNe 12 ForestsOpEvBr 13 ForestsClEvBr 14 ForestsDeEvBr 15 ForestsOpEvNe

16 ForestsClEvNe 17 ForestsDeEvNe 18 WetlandMangro 19 WetlandSwamps 20 WetlandMarshl

21 WaterBodyMari 22 WaterBodyCont 23 PermanentSnow 24 CropSeasWater 25 CropCereaIrri

26 CropCereaRain 27 CropBroadIrri 28 CropBroadRain 29 UrbanBlUpArea

Figure 4.5: Samples of images for each one of the 29 Land-Use and Land-Cover (LULC) classes contained in the Sentinel2GlobalLULC dataset

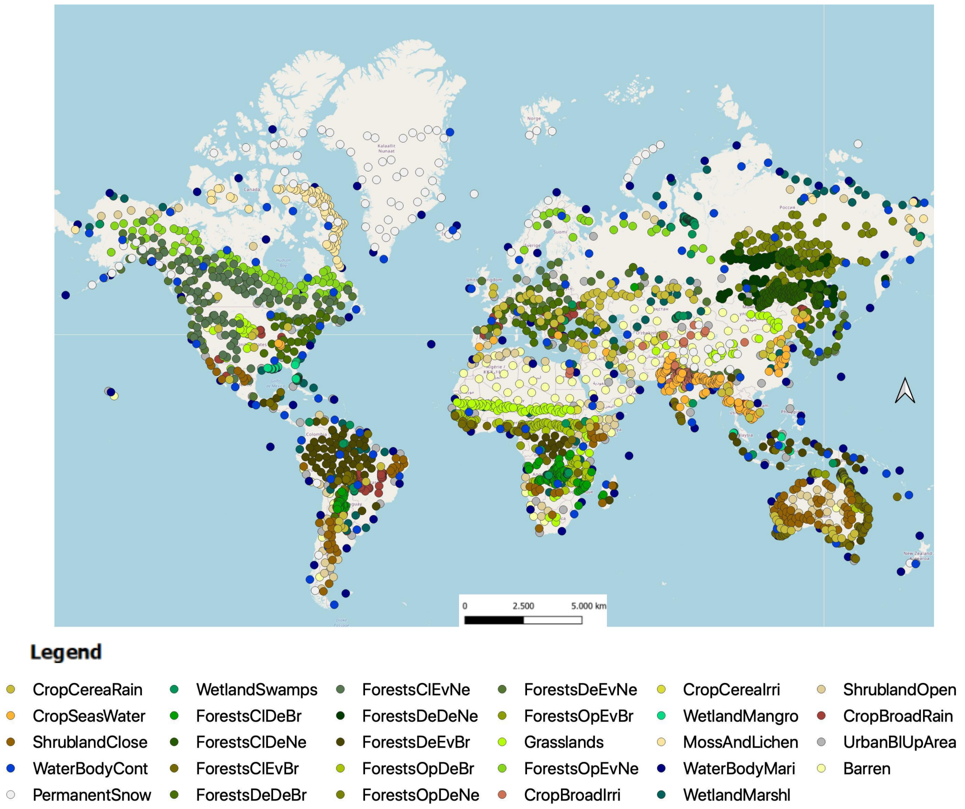


Figure 4.6: Global distribution of the selected 100 images for each Land-Use and Land-Cover (LULC) class to perform the validation of the 29 LULC classes contained in the Sentinel2GlobalLULC dataset. An add-hoc script in R was used to maximize the geographical distance among the 100 points of each class.

CHAPTER 5

A first study exploring the performance of deep learning CNNs in Sentinel2GlobalLULC classification for global land use/cover mapping

Contents

This chapter is to be submitted as a paper to a Q1 journal.

5.1 Introduction and motivation

Let us first give a detailed introduction to the main topic of the present chapter. As we introduced in the last chapter remote sensing data and especially data related to LULC mapping of our planet is continuously growing in parallel to new methods and models to exploit all these data. Now that all these ingredients are available, the research community is trying to build more robust automatic LULC mapping systems using new high performance DL models. In fact, the choice of DL comes from the outstanding results achieved by the latter in most computer vision tasks and especially in image classification; in addition to the fact the current abundance in remote sensing images. Particularly, Sentinel2LULC which to our knowledge is the largest, free, global and high resolution RGB images dataset. All these characteristics represents a real source gold for DL models. Hence, in this chapter we will explore for the first time: the performance of our deep-learning-ready Sentinel2LULC dataset using various CNN models with the aim to build a robust global LULC mapping model. During this first analysis we will not apply any additional preprocessing methods, and this assuming that all the preprocessing that our dataset has undergone during its creation to make it a high quality product is sufficient for DL models training. The exploitation of various CNN models and the analysis of their results allowed us to expose all learnt lessons and conclusions that would serve to establish our future works and improve the achieved results. In the first part of this chapter, we will report related works that tried to build an automatic LULC mapping system using other remote sensing datasets. Then, we will present the used methodology in our DL analysis in conjunction with Sentinel2LULC dataset. Afterwards, we will expose the achieved results of this experimental study along with a geographic representation and analysis of these results. Finally, we will present what has been concluded from this study.

5.2 Related works

According to the Food and Agriculture Organisation (FAO) Land mapping in earth observation is divided into two main categories: land cover and land use [170]. Land cover is the biophysical cover on Earth's surface, whereas land use represents the way Earth's surface is exploited, maintained or changed by humans. Land use and land cover are strongly related, and their joint classification is almost inevitable. For this reason, in most related works as it is the case in our dataset, land use and land cover (LULC) classification is considered as a whole and not separated concepts. The differences in related works consist more in their geographic coverage as being global or local, their classification schema that represent the number of considred LULC classes, the used data type and the adopted DL models. To summarize all these researches, in this section we will report all DL based LULC mapping related works in regards to all these specifications. In fact, The number of papers dealing with DL applications into land cover and land use classification is doubling each year since 2015, and for a complete overview we recommend the reader to consult the following paper summarizing all these studies [186].

Firstly, authors in [111] explored the performance of an ImageNet pre-trained CNN with an aerial UC Merced dataset for LULC mapping. This study focused more on the evaluation of different scenarios for the CNN fine-tuning and the importance of the pre-trained features from ImageNet in this task. Regarding the CNN model choice, they opted for Overfeat which is an improved version of AlexNet. Whereas, in [206], the authors evaluated the performance of a novel CNN model called ASPP-Unet and ResASPP-Unet in urban land use classification over the city of Beijing in China. In fact, these models were built from the well known U-Net by integrating the ASPP approach to the latter. To reach their objective, they trained these CNNs from scratch and tested them on WorldView-2 (WV2) and WorldView-3 (WV3) imagery over the area of interest. Then, authors in [73], explored the problem of automatic LULC mapping using DL in conjunction with five different LULC mapping datasets including a new one that they proposed in their study. this new dataset they proposed consists of 10 various classes with a total of 27,000 labeled Sentinel-2 satellite images. As a DL evaluation they explored the performance of two different CNNs (ResNet-50 and GoogLeNet). To our knowledge, the only work that proposed a global LULC map is [88]. The study in this work was based on a novel and large dataset of 24,000 5km x 5km sentinel-2 image at 1Om resolution that were hand labeled giving ten classes (water, trees, grass, flooded, vegetation, crops, scrub/shrub, built area, bare ground, snow/ice, and clouds).

Related work	Evaluated dataset	Source	Nbr classes	Total Nbr Images	Geographic scale	Evaluated DL models	Pretrained on	Best results (%)
[72]	Eurosat	Sentinel-2	10	27000	Local	GoogLeNet and ResNet-50	ImageNet	98.57 (ResNet-50)
[72]	UC Merced Land Use	USGS National Geospatial Program	21	21000	Local	GoogLeNet and ResNet-51	ImageNet	97.32 (GoogLeNet)
[72]	AID	Multi-source (Google Earth Imagery)	30	10000	Local (Limited number of countries)	GoogLeNet and ResNet-52	ImageNet	94.38 (ResNet-50)
[72]	SAT-6	NAIP National Agriculture Imagery Program	6	450000	Local	GoogLeNet and ResNet-53	ImageNet	99.56 (ResNet-50)
[72]	BCS	Spot sensor	2	37015	Local (Limited number of countries)	GoogLeNet and ResNet-54	ImageNet	93.57 (ResNet-50)
[111]	UC Merced Land Use	Arial	21	2100	Local	Overfeat(AlexNet)	ImageNet	92.4
[206]	WorldView2 imagery and WorldView3 imagery	Maxar	6	NA	Local	New developped models(ASPP-Unet and ResASPP-Unet)	None	ResASPP-Unet: 87.1(WV2) 84(WV3)
[88]	private dataset	Sentinel-2	10	24000	Global	UNet	None	85

5.3 Methodology

In this section, we present the adopted methodology to explore the performance of 12 different CNNs trained on a subset of our Sentinel2LULC dataset. The used methodology is composed of three main steps: First, we will describe the characteristics of the new balanced subset selected from Sentinel2LULC and the motivation behind this selection. Second, we will present the used CNN models and describe the followed experimental setup to train them on Sentinel2LULC images.

5.3.1 Sentinel2GlobalLULC subset selection

In general, a skewed data distribution arises naturally in many applications where some class occurs with reduced frequency in comparison to the other classes. Particularly, in such a complex and global setting, as in Sentinel2GlobalLULC dataset, the global distribution varies from one LULC class to another. This fact makes that Sentinel2LULC with 29 LULC classes is inherently unbalanced and the number of images in LULC classes varies from few hundreds to several thousands depending on the abundance of each class around the world. Many studies in the literature explored the impact of class imbalance on the outcome performance. Notably, in [86], the authors have found that the classifier sensitivity to the data imbalance increases as the problem complexity increases. Hence, giving the complexity of our classification problem, we know that this class unbalance could seriously affect the performance of the evaluated classifier on our dataset. Therefore, we elaborated a downsampling approach with a data balancing algorithm that selects 354 images from each LULC class. In fact, this number equals exactly the number of images contained in the smallest LULC class. The adopted selection algorithm was based on the center coordinates (longitude and latitude) of each image and it uses them to guarantee an evenly distributed and large geographic representation for each LULC class. In summary, the created subset after this selection winded up having 10.266 images carefully selected from Sentinel2LULC and distributed into 29 LULC classes where each class has exactly 354 images carefully annotated and well distributed around the world.

5.3.2 Deep learning models training

After building the new subset containing 10.266 images carefully extracted from the original Sentinel2LULC, we trained 12 different CNNs on the resulted subset of Sentinel2LULC dataset. All analyzed CNNs were fine tuned on our Sentinel2LULC subset using their pre-trained version on ImageNet. No preprocessing methods were performed. The used subset was split into a training set of 70% and a test set of 30% evenly distributed on all LULC classes. All CNNs were trained using a $batchsize = 8$ and a $learningrate = 0.03$ during 100 epochs. We have used the following 12 different CNN models:

- InceptionV3
- VGG16
- VGG19

- ResNet50
- Xception
- InceptionResNetV2
- MobileNet
- DenseNet121
- DenseNet169
- DenseNet201
- NASNetLarge
- NASNetMobile

5.4 Experimental results

5.4.1 Overall accuracy results

In this subsection, we will present the results achieved by each one of the 12 evaluated CNNs. Therefore, we will explore for each CNN, the overall accuracy which is calculated in the following manner:

$$\text{Overall accuracy} = \frac{\text{Correctly classified test images for all LULC classes}}{\text{All images in the test set}}$$

Trained CNN	Overall Accuracy (%)
InceptionV3	68.28875281985176
VGG16	63.55140186915887
VGG19	56.07476635514018
ResNet50	19.497260715436673
Xception	72.60715436674187
InceptionResNetV2	72.51047373509507
MobileNet	75,70093457943925
DenseNet121	82.53303254914599
DenseNet169	82.04962939091202
DenseNet201	80.59941991621012
NASNetLarge	75.81647530259264
NASNetMobile	73.25169191105382

As you can see in the table above, the three DenseNet CNNs variants (DenseNet121, DenseNet169, DenseNet201) were the only ones to exceed 80% in the overall accuracy. Thus in the following subsections, we will focus on highlighting their results in more details.

5.4.2 Exploration in details of the three best performing CNNs

In this subsection, we will focus and expose in detail the results achieved by the three DenseNet variants (DenseNet121, DenseNet169, DenseNet201) that were the only CNNs among the 12 explored ones to achieve an overall accuracy higher than 80%. Therefore, we will present for each one of the three CNNs regrading each LULC class: the number of True Positive (TP), the number of False Negative (FN), the number of True Negative (TN), the number of False Positive (FP). In fact, computing these numbers for each LULC class will help us find the reached Accuracy(%), Recall(%), Precision(%) and F1-score(%) for each one of the 29 LULC classes.

These used metrics were calculated in the following manner:

For each LULC class C:

$$TP = \text{the number of correctly classified images as C}$$

$$TN = \text{the number of correctly classified images as other LULC class while they belong to C}$$

$$FN = \text{the number of misclassified images as other LULC class while they belong to C}$$

$$FP = \text{the number of misclassified images as C while they belong to other LULC classes}$$

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)}$$

$$Precision = \frac{(TP)}{(TP + FP)}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F_1score = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

As we can see in the results presented in the tables below (Table5.1, Table5.2, Table5.3), for the three DensNet models (DenseNet121, DenseNet169, DenseNet201), most LULC classes achieved an F1-score higher than 80% except (WetlandMarshl, WaterBodyCont, ForestsOpEvNe, CropCereaRain, ForestsClEvNe) for DenseNet121, these classes (WetlandMarshl, WaterBodyCont, ForestsOpEvNe, CropCereaRain, ForestsClEvBr, ForestsClEvNe) for DenseNet169 and the following classes (WetlandMarshl, ForestsOpDeBr, ForestsOpEvNe, ForestsDeDeNe, ForestsOpDeNe, WaterBodyMari, CropCereaRain, ForestsClEvBr, ForestsClDeNe, ForestsClEvNe) for DenseNet201. These LULC classes in each CNN model are the ones who has achieved the lowest accuracy too and by consequence DensNet201 who has 10 classes with an F1-score below 80% was the one with the lowest overall accuracy. From these results, we can also notice that there are three LULC classes (WetlandMarshl, CropCereaRain, ForestsClEvNe) that were difficult to classify correctly for all the evaluated DensNet variants. In fact, we believe that these three classes has a larger geographic distribution which induce a high variability in the visual features of their images and by consequence they are more difficult to detect for the trained CNNs. A proposition that we will explore in future works to overcome issues related to these three classes and improve our CNN accuracy: is to elaborate an additional preprocessing step to reduce their data variability and normalize their visual characteristics.

5.4.2.1 Exploration in details of DenseNet121 results

LULC_Class	F1-score	TP	FP	TN	FN	Accuracy	Recall	Precision
Class_18_WetlandMangro	92.82297	194	10	2986	20	99.06542	90.65421	95.09804
Class_12_ForestsOpEvBr	88.37209	190	26	2970	24	98.44237	88.78505	87.96296
Class_23_PermanentSnow	98.8345	212	3	2993	2	99.84424	99.06542	98.60465
Class_20_WetlandMarshl	**69.97389**	**134**	**35**	**2961**	**80**	**96.41745**	**62.61682**	**79.28994**
Class_8_ForestsDeDeBr	90.16787	188	15	2981	26	98.72274	87.85047	92.61084
Class_4_ShrublandOpen	91.78744	190	10	2986	24	98.94081	88.78505	95
Class_22_WaterBodyCont	**79.38144**	**154**	**20**	**2976**	**60**	**97.50779**	**71.96262**	**88.50575**
Class_6_ForestsOpDeBr	86.95652	180	20	2976	34	98.31776	84.11215	90
Class_5_SrublandClose	89.85507	186	14	2982	28	98.69159	86.91589	93
Class_2_MossAndLichen	95.962	202	5	2991	12	99.4704	94.39252	97.58454
Class_15_ForestsOpEvNe	**77.19298**	**154**	**31**	**2965**	**60**	**97.16511**	**71.96262**	**83.24324**
Class_11_ForestsDeDeNe	80.59701	162	26	2970	52	97.57009	75.70093	86.17021
Class_9_ForestsOpDeNe	82.35294	168	26	2970	46	97.75701	78.50467	86.59794
Class_21_WaterBodyMari	83.29298	172	27	2969	42	97.85047	80.37383	86.43216
Class_25_CropCereaIrri	85.85859	170	12	2984	44	98.25545	79.43925	93.40659
Class_17_ForestsDeEvNe	**78.125**	**150**	**20**	**2976**	**64**	**97.38318**	**70.09346**	**88.23529**
Class_24_CropSeasWater	86.33094	180	23	2973	34	98.2243	84.11215	88.66995
Class_27_CropBroadIrri	91.74312	200	22	2974	14	98.8785	93.45794	90.09009
Class_26_CropCereaRain	**79.60199**	**160**	**28**	**2968**	**54**	**97.44548**	**74.76636**	**85.10638**
Class_13_ForestsClEvBr	80.59701	162	26	2970	52	97.57009	75.70093	86.17021
Class_14_ForestsDeEvBr	96.71362	206	6	2990	8	99.56386	96.26168	97.16981
Class_28_CropBroadRain	86.91358	176	15	2981	38	98.34891	82.24299	92.1466
Class_10_ForestsClDeNe	80.59701	162	26	2970	52	97.57009	75.70093	86.17021
Class_1_BarrenLands	93.26923	194	8	2988	20	99.12773	90.65421	96.0396
Class_29_UrbanBlUpArea	98.82353	210	1	2995	4	99.84424	98.13084	99.52607
Class_7_ForestsClDeBr	87.65743	174	9	2987	40	98.47352	81.30841	95.08197
Class_16_ForestsClEvNe	**73.09645**	144	36	2960	70	96.69782	67.28972	80
Class_3_Grasslands	88.56448	182	15	2981	32	98.53583	85.04673	92.38579
Class_19_WetlandSwamps	83.90244	172	24	2972	42	97.94393	80.37383	87.7551

Table 5.1: Experimental results of DenseNet121 exposed in details

5.4.2.2 Exploration in details of DenseNet169 results

LULC_Class	F1-score	TP	FP	TN	FN	Accuracy	Recall	Precision
Class_18_WetlandMangro	89.95215	188	16	2980	26	98.69159	87.85047	92.15686
Class_12_ForestsOpEvBr	88.47059	188	23	2973	26	98.47352	87.85047	89.09953
Class_23_PermanentSnow	97.9021	210	5	2991	4	99.71963	98.13084	97.67442
Class_20_WetlandMarshl	**67.88512**	**130**	**39**	**2957**	**84**	**96.16822**	**60.74766**	**76.92308**
Class_8_ForestsDeDeBr	87.37864	180	18	2978	34	98.38006	84.11215	90.90909
Class_4_ShrublandOpen	91.56627	190	11	2985	24	98.90966	88.78505	94.52736
Class_22_WaterBodyCont	**77.86667**	**146**	**15**	**2981**	**68**	**97.41433**	**68.2243**	**90.68323**
Class_6_ForestsOpDeBr	82.93963	158	9	2987	56	97.97508	73.83178	94.61078
Class_5_SrublandClose	86.13861	174	16	2980	40	98.25545	81.30841	91.57895
Class_2_MossAndLichen	94.73684	198	6	2990	16	99.31464	92.52336	97.05882
Class_15_ForestsOpEvNe	**74.80519**	**144**	**27**	**2969**	**70**	**96.97819**	**67.28972**	**84.21053**
Class_11_ForestsDeDeNe	81.38958	164	25	2971	50	97.66355	76.63551	86.77249
Class_9_ForestsOpDeNe	82.17822	166	24	2972	48	97.75701	77.57009	87.36842
Class_21_WaterBodyMari	83.41232	176	32	2964	38	97.81931	82.24299	84.61538
Class_25_CropCereaIrri	88.66499	176	7	2989	38	98.59813	82.24299	96.17486
Class_17_ForestsDeEvNe	81.92771	170	31	2965	44	97.66355	79.43925	84.57711
Class_24_CropSeasWater	86.13861	174	16	2980	40	98.25545	81.30841	91.57895
Class_27_CropBroadIrri	88.78505	190	24	2972	24	98.50467	88.78505	88.78505
Class_26_CropCereaRain	**75.91241**	**156**	**41**	**2955**	**58**	**96.91589**	**72.8972**	**79.18782**
Class_13_ForestsClEvBr	**78.78788**	**156**	**26**	**2970**	**58**	**97.38318**	**72.8972**	**85.71429**
Class_14_ForestsDeEvBr	97.16981	206	4	2992	8	99.62617	96.26168	98.09524
Class_28_CropBroadRain	85.21303	170	15	2981	44	98.16199	79.43925	91.89189
Class_10_ForestsClDeNe	84.50704	180	32	2964	34	97.94393	84.11215	84.90566
Class_1_BarrenLands	92.23301	190	8	2988	24	99.00312	88.78505	95.9596
Class_29_UrbanBlUpArea	99.06542	212	2	2994	2	99.87539	99.06542	99.06542
Class_7_ForestsClDeBr	89.81481	194	24	2972	20	98.62928	90.65421	88.99083
Class_16_ForestsClEvNe	**67.72487**	**128**	**36**	**2960**	**86**	**96.19938**	**59.81308**	**78.04878**
Class_3_Grasslands	88.78049	182	14	2982	32	98.56698	85.04673	92.85714
Class_19_WetlandSwamps	85.64477	176	21	2975	38	98.16199	82.24299	89.3401

Table 5.2: Experimental results of DenseNet169 exposed in details

5.4.2.3 Exploration in details of DenseNet201 results

LULC_Class	F1-score	TP	FP	TN	FN	Accuracy	Recall	Precision
Class_18_WetlandMangro	91.80328	196	17	2979	18	98.90966	91.58879	92.01878
Class_12_ForestsOpEvBr	89.83452	190	19	2977	24	98.66044	88.78505	90.90909
Class_23_PermanentSnow	98.36066	210	3	2993	4	99.78193	98.13084	98.59155
Class_20_WetlandMarshl	**67.92453**	**126**	**31**	**2965**	**88**	**96.29283**	**58.8785**	**80.25478**
Class_8_ForestsDeDeBr	90.56604	192	18	2978	22	98.75389	89.71963	91.42857
Class_4_ShrublandOpen	92.82297	194	10	2986	20	99.06542	90.65421	95.09804
Class_22_WaterBodyCont	80.20305	158	22	2974	56	97.57009	73.83178	87.77778
Class_6_ForestsOpDeBr	**79.59698**	**158**	**25**	**2971**	**56**	**97.47664**	**73.83178**	**86.3388**
Class_5_SrublandClose	89.7561	184	12	2984	30	98.69159	85.98131	93.87755
Class_2_MossAndLichen	96.68246	204	4	2992	10	99.56386	95.3271	98.07692
Class_15_ForestsOpEvNe	**74.4186**	**144**	**29**	**2967**	**70**	**96.91589**	**67.28972**	**83.23699**
Class_11_ForestsDeDeNe	**78.84615**	**164**	**38**	**2958**	**50**	**97.25857**	**76.63551**	**81.18812**
Class_9_ForestsOpDeNe	**79.03614**	**164**	**37**	**2959**	**50**	**97.28972**	**76.63551**	**81.59204**
Class_21_WaterBodyMari	**79.02439**	**162**	**34**	**2962**	**52**	**97.32087**	**75.70093**	**82.65306**
Class_25_CropCereaIrri	86	172	14	2982	42	98.25545	80.37383	92.47312
Class_17_ForestsDeEvNe	83.90244	172	24	2972	42	97.94393	80.37383	87.7551
Class_24_CropSeasWater	85.08557	174	21	2975	40	98.09969	81.30841	89.23077
Class_27_CropBroadIrri	89.41176	190	21	2975	24	98.59813	88.78505	90.04739
Class_26_CropCereaRain	**72.44094**	**138**	**29**	**2967**	**76**	**96.72897**	**64.48598**	**82.63473**
Class_13_ForestsClEvBr	**74.03599**	**144**	**31**	**2965**	**70**	**96.85358**	**67.28972**	**82.28571**
Class_14_ForestsDeEvBr	97.65258	208	4	2992	6	99.68847	97.19626	98.11321
Class_28_CropBroadRain	82.494	172	31	2965	42	97.72586	80.37383	84.72906
Class_10_ForestsClDeNe	**75.62189**	**152**	**36**	**2960**	**62**	**96.94704**	**71.02804**	**80.85106**
Class_1_BarrenLands	94.45783	196	5	2991	18	99.28349	91.58879	97.51244
Class_29_UrbanBlUpArea	98.34515	208	1	2995	6	99.78193	97.19626	99.52153
Class_7_ForestsClDeBr	85.3598	172	17	2979	42	98.16199	80.37383	91.00529
Class_16_ForestsClEvNe	**59.13043**	**102**	**29**	**2967**	**112**	**95.60748**	**47.66355**	**77.8626**
Class_3_Grasslands	87.29017	182	21	2975	32	98.34891	85.04673	89.65517
Class_19_WetlandSwamps	83.95062	170	21	2975	44	97.97508	79.43925	89.00524

Table 5.3: Experimental results of DenseNet201 exposed in details

5.4.3 Geographic exploration of the best performing CNN

In this section, we will focus on the best performing CNN in our analysis which is DenseNet-121 and highlight visually its results in a global map. This analysis helped us to have more insights on the CNN behaviour from a visual standpoint and see how this CNN learns to classify each one of the 29 LULC classes according to the geographic distribution of their training and test data points. This visualization was carried out using geographic coordinates of each data record in the training and test sets. We presented the 29 figures of this elaborated analysis in the Appendix. Each one of the 29 figures, represents the geographic distribution of the training and test images around the world for each one of the 29 LULC classes, and for the test images we highlighted separately the correctly classified and misclassified ones. In fact, the green points are training images used for that class, while pistachio green and grey point correspond to the correctly classified and misclassified images respectively after test for that LULC class. Each point in the maps was created using the center coordinates of its corresponding image.

As we can see in most LULC classes, the distribution of correctly classified points follows the training one, and most correctly classified images are the one situated within the geographic cluster formed by the training points in that LULC class. In fact most misclassified images are outliers situated away or detached from the main training body. This observation allow us to establish a n idea about the geographic behavior of the trained CNN. In fact, we can conclude that the geographic position of a certain point is very deterministic in its classification and the explanation of this fact is that images of the same LULC class that are geographically close to each others are more prone to have similar visual features. Particularly, the geographic distribution in (WetlandMarshl, WaterBodyCont, ForestsOpEvNe, CropCereaRain, ForestsClEvNe) classes that causes the decrease in the overall accracy for DenseNet121 CNN, containes various clusters; and this fact could be the hint behind their classification results. Thus, a possible outlook for future improvement of the actual best analysed CNN model is to apply a normalization preprocessing step to reduce data variability in these classes. Furthermore, a training strategy that could help take this geographic distribution factor into consideration is to include the center coordinates (longitude and latitude) of used images as additional input during the training and the test phases of the CNN model in a multi-input training scenario.

5.5 Conclusion

In this chapter, we explored for the first time the performance of DL models on our dataset Sentinel2GlobalLULC for global LULC mapping. In fact, we evaluated 12 different CNNs and reported the overall accuracy reached in each one of them. Then, we presented the results of the three best performing CNNs among them with more metrics and exposed the number of TP, FP, TN, FN in each one to conclude the F1-score of these models regarding each LULC class. In this experimental, DensNet variants (DenseNet121, DenseNet169, DenseNet201) have shown to be the most efficient as they were the only ones that exceeded an overall accuracy of 80%. Afterwards, we dived deeper in the results of the most efficient CNN of these three variants which is DenseNet121. In fact, we provided 29 figures to vizualize the learning behaviour of this CNN for each LULC class. In each one of these figures we represented the geographic distribution in the world map of the training images, the correctly classified and misclassified images in a given LULC class. This analysis has allowed us to examine the effect of data variability in images from different locations in the world on the test results. We concluded that a normalization preprocessing phase might bring an improvement to the classification accuracy since most classes that caused a drop in accuracy are the one where training and test set are distributed over different areas of the world. Another interesting conclusion that will be a part of our future work regarding Sentinel2GlobalLULC dataset is the inclusion of each image coordinates (longitude and latitude) as an additional input to train the used CNN for this complex task.

CHAPTER 6

Conclusions & future works

This thesis was devoted principally to evaluation our proposed data preprocessing approaches with the aim to train DL models for two complex image classification tasks: biomedical images classification for automatic breast cancer diagnosis and satellite images classification for global LULC mapping.

In the first application, we started by elaborating an overview analysis of all related works to to the benchmark dataset BreakHis. Then, we combined all learnt lessons from this analysis, to design an ideal system for breast cancer automatic diagnosis from both clinical and technical standpoints. In fact, the system we designed was built from the best reformulation for BreakHis dataset classification problem in conjunction with the most adequate preprocessing methods, the best performing CNN in this task and the appropriate learning strategies. After the experimental implementation of all these ingredients together, we didn't reach what we were hypothetically expecting from such a system that constitutes a combination of the best performing methods and models in the literature at pre- and pos-processing levels for this problem. We concluded from these results that the complexity of this problem is mostly related to its data quality and annotation, and adopting such a dataset in a very complex classification problem hugely affect the performance of the trained DL model even in a well built methodological approach with very efficient data preprocessing techniques.

To avoid these data quality related issues, we built our own dataset in the second application of this thesis. The latter consists in satellite images classification for global LULC mapping. The dataset we created called Sentinel2LULC is the largest global high resolution and free satellite images dataset for this purpose. During its creation we combined all global remote sensing data available in the literature for this problem. We consider that the way Sentinel2LULC was created itself consists in the data preprocessing and the contribution we brought to this problem as we were having during this process a full control on the annotation quality and classification structure and we made sure that it is the most suitable for DL models training. Then, we evaluated the potential of Sentinel2LULC for global LULC mapping using various CNN models. The elaborated experimental analysis achieved very promising results and confirmed the the high quality of our dataset.

These two applications explored in this thesis under different data quality circumstances has clearly showed us that one should be aware of the most important point which is the data quality he uses and its suitability for DL training when dealing with very complex CV tasks such as those addressed in this thesis. In fact, the data itself, its quality and its characteristics remains the key element that decides either the DL model will reach a good performance or not.

As future works, and based on the learnt lessons in this thesis, we will give below a list of possible extensions related to each application of this thesis that we will address within a Postdoctoral work:

The possible future work planned for the first application presented in chapters 2 and 3 would be : After BreakHis analysis within the ideal reformulation for this problem MIM and after the experimental evaluation of this reformulation using the most adequate DL models, pre- and post processing methods; we have concluded that to achieve a good performance for MIM classification we should have a better quality dataset made specially for this task. Thus, a very promising future work in this research line would be to take all the necessary time, expenses and medical expertise to create a new histopathological imagery dataset that will fit all these requirements to establish an ideal breast cancer CAD system for this problem as we formulated in our analysis.

The possible future work planned for the second application presented in chapters 4 and 5 would be: After the creation of the high quality Sentinel2GlobalLULC dataset and exploring its potential for the first time in global LULC mapping using DL models, we deduced many promising future works. First, we think that applying additional preprocessing phase like image normalization for samples collected from different locations in the world might bring an improvement to the classification accuracy and overcome the data variability issue. Second, since we have discovered that there is a possible correlation between the geographic coordinates and the LULC class in the images of our dataset, an interesting future work would be the inclusion of each image coordinates (longitude and latitude) as an additional input to train DL models for this complex task.

Publications included in this thesis

Journal publications:

- [15] Benhammou, Y., Achchab, B., Herrera, F., Tabik, S. (2020). BreakHis based breast cancer automatic diagnosis using deep learning: Taxonomy, survey and insights. **Published in Neurocomputing**, 375, 9-24..
- [16] Benhammou, Y., Alcaraz-Segura, D., Guirado, E., Khaldi, R., Achchab, B., Herrera, F., Tabik, S. (2021). Sentinel2GlobalLULC: A deep-learning-ready Sentinel-2 RGB image dataset for global land use/cover mapping. **Under Review in Scientific Data-Nature Publishing Group** , and **Published** as a preprint in BioRxiv.

Conference Proceedings:

- [17] Benhammou, Y., Tabik, S., Achchab, B., Herrera, F. (2018, May). A first study exploring the performance of the state-of-the art CNN model in the problem of breast cancer. **Presented during LOPAL'18 international conference** , and **Published In Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications** (pp. 1-6).

Other collaborations realized during this thesis

Journal publications:

- [137] Pérez-Hernández, F., Rodríguez-Ortega, J., Benhammou, Y., Herrera, F., Tabik, S. (2021). CI-dataset and DetDSCI methodology for detecting too small and too large critical infrastructures in satellite images: Airports and electrical substations as case study. **Published in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, 14, 12149-12162.

- [91] Khaldi, R., Alcaraz-Segura, D., Guirado, E., Benhammou, Y., El Afia, A., Herrera, F., and Tabik, S.: TimeSpec4LULC: A Global Deep Learning-driven Dataset of MODIS Terra-Aqua Multi-Spectral Time Series for LULC Mapping and Change Detection. **Under review in Earth System Science Data**, in review, 2021.

Book chapters:

- [112] Martorell-Marugán, J., Tabik, S., Benhammou, Y., del Val, C., Zwir, I., Herrera, F., Carmona-Sáez, P. (2019). Deep learning in omics data analysis and precision medicine. **Published in Exon Publications**, 37-53.

APPENDIX A

Annexe

A.1 Appendix: Geographic exploration of DenseNet-121 results on Sentinel2LULC

This section is an appendix to Chapter 5. In this section we will vizualize results of DenseNet-121 in Sentinel2LULC classification in a global map. This visualization was carried out using geographic coordinates of each data record in the training and test sets. Each one of the 29 figures presented below represents the geographic distribution of the training, the correctly classified and misclassified images for each one of the 29 LULC classes. Green points are training records used for that class, while pistachio green and grey point correspond to correctly classified and misclassified images respectively. Each point in the maps was created using the center coordinates of its corresponding image.

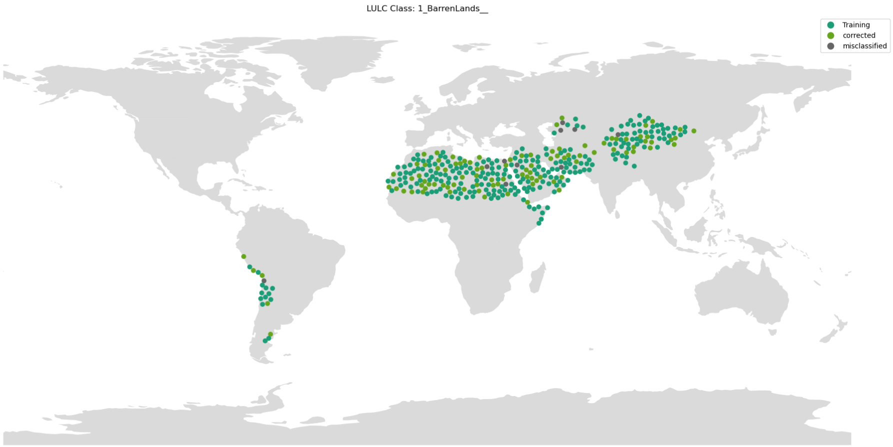


Figure A.1: Geographic distribution of DenseNet-121 results of BarrenLands class

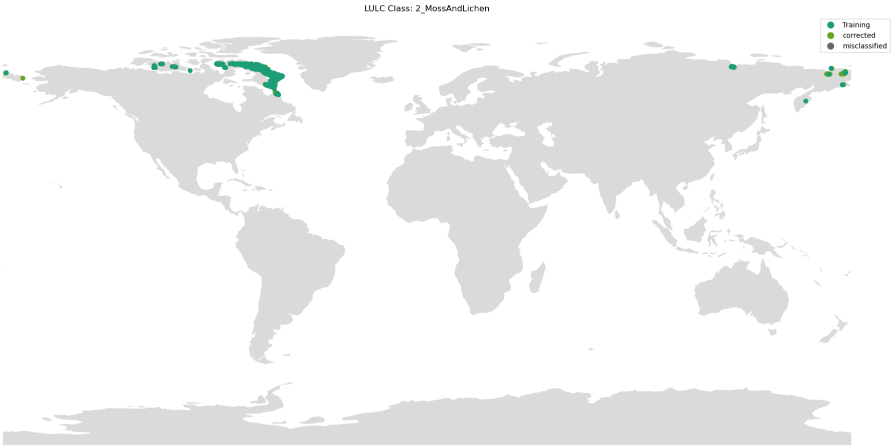


Figure A.2: Geographic distribution of DenseNet-121 results of MossAndLichen class

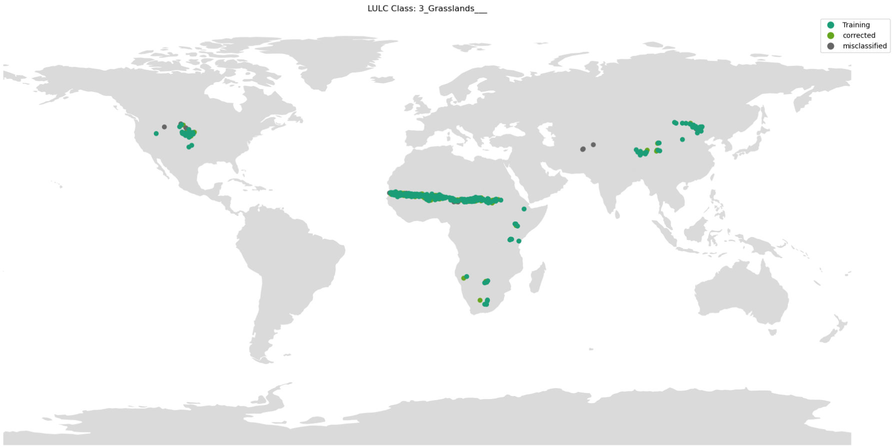


Figure A.3: Geographic distribution of DenseNet-121 results of Grasslands class

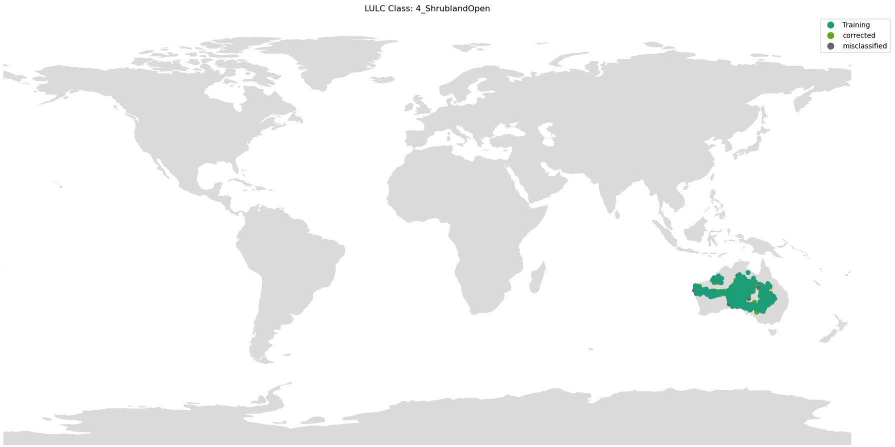


Figure A.4: Geographic distribution of DenseNet-121 results of ShrublandOpen class

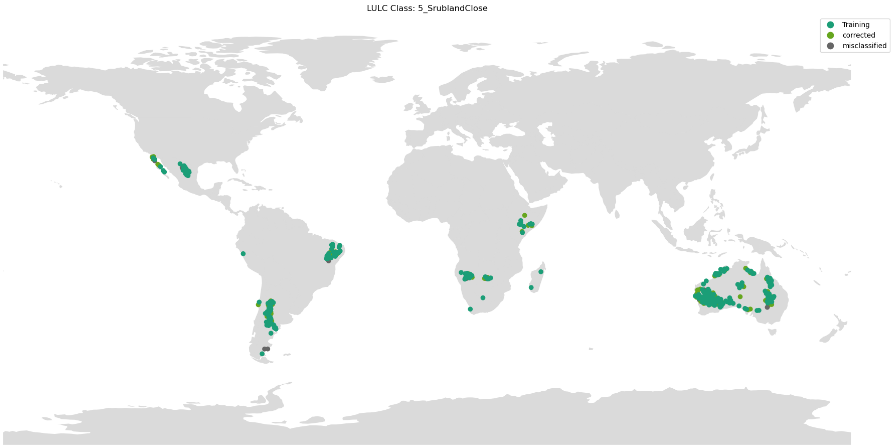


Figure A.5: Geographic distribution of DenseNet-121 results of SrublandClose class

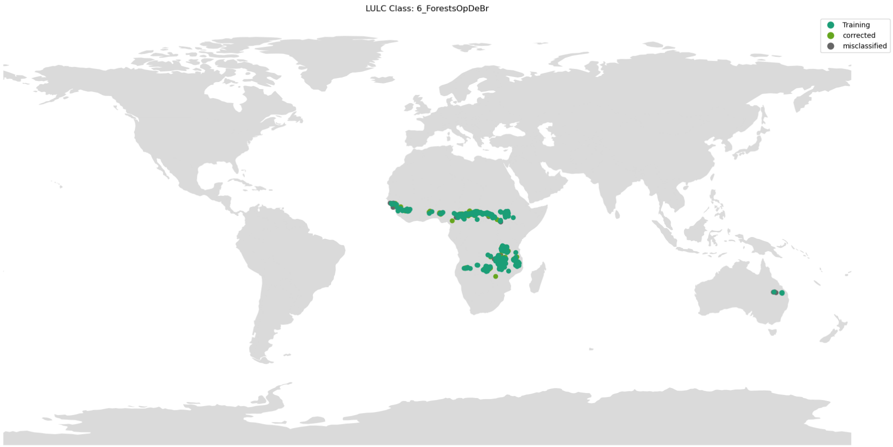


Figure A.6: Geographic distribution of DenseNet-121 results of ForestsOpDeBr class

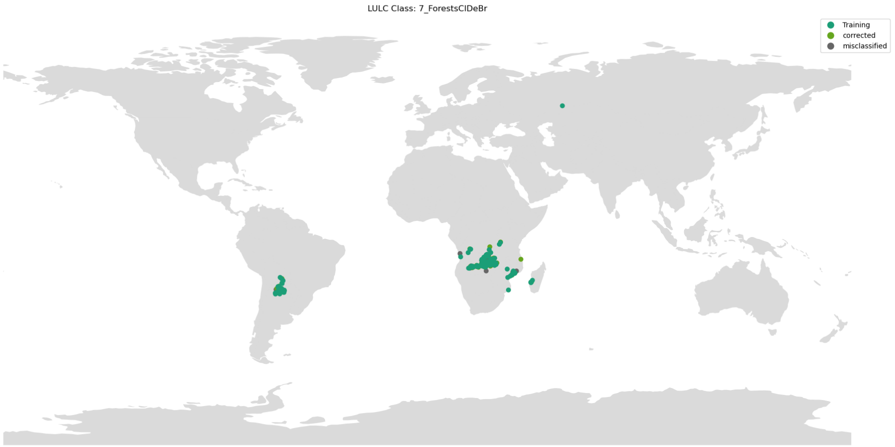


Figure A.7: Geographic distribution of DenseNet-121 results of ForestsClDeBr class

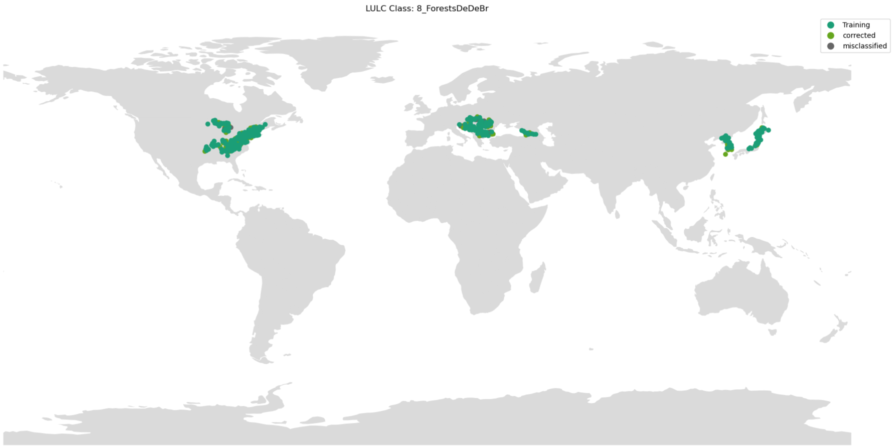


Figure A.8: Geographic distribution of DenseNet-121 results of ForestsDeDeBr class

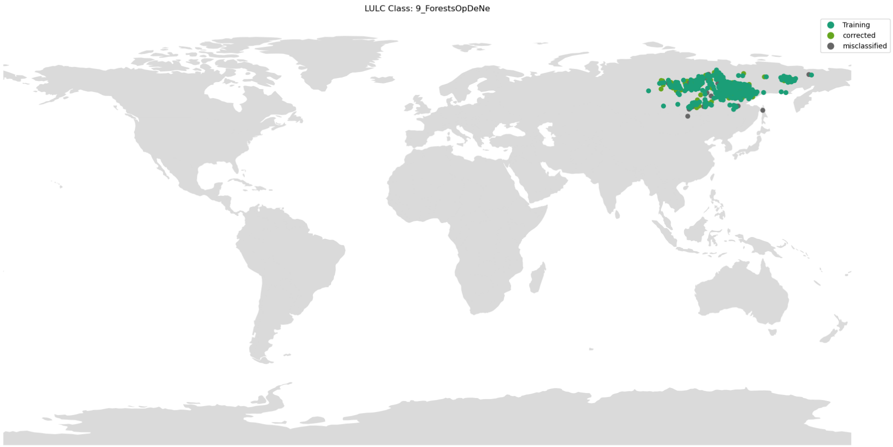


Figure A.9: Geographic distribution of DenseNet-121 results of ForestsOpDeNe class

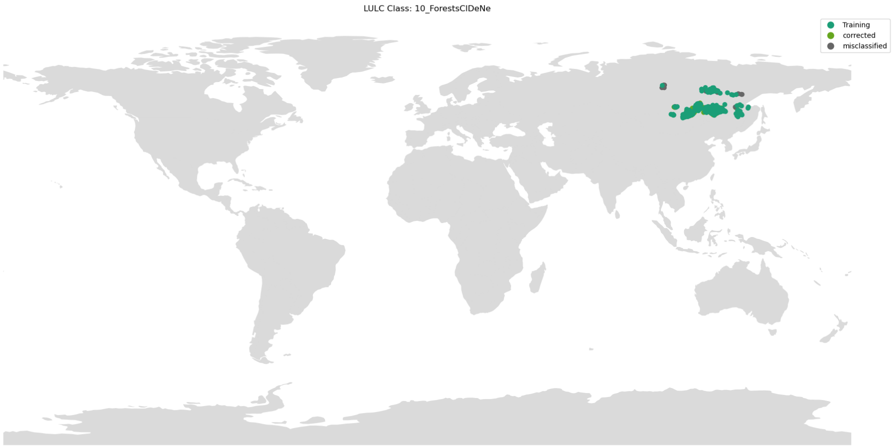


Figure A.10: Geographic distribution of DenseNet-121 results of ForestsClDeNe class

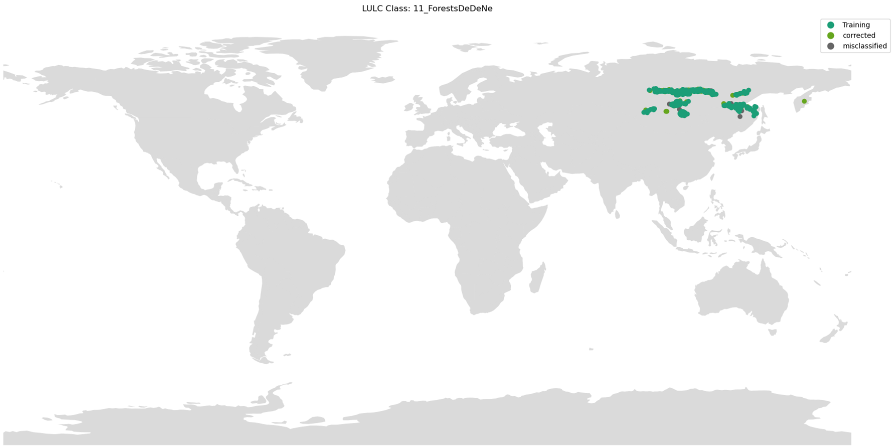


Figure A.11: Geographic distribution of DenseNet-121 results of ForestsDeDeNe class

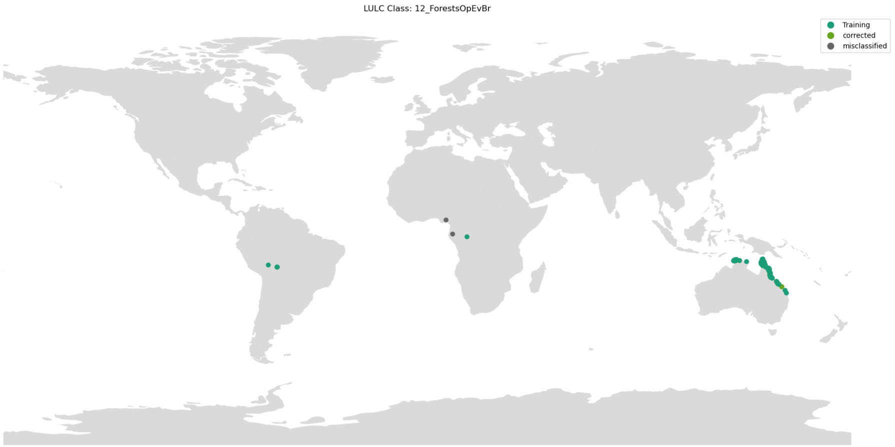


Figure A.12: Geographic distribution of DenseNet-121 results of ForestsOpEvBr class

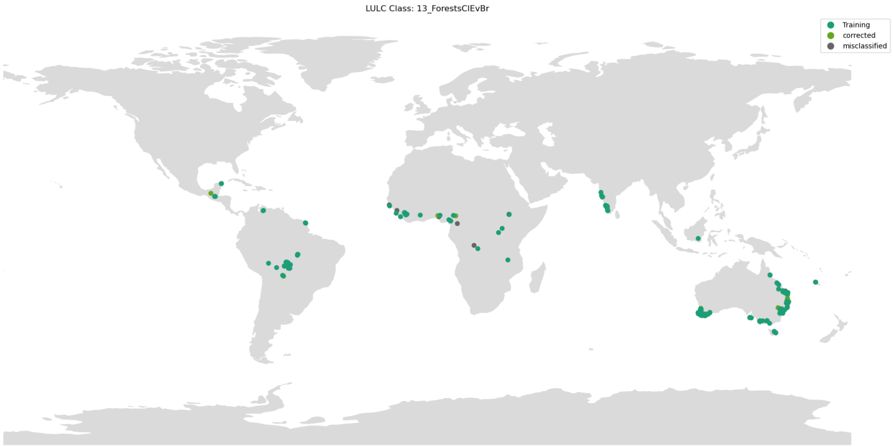


Figure A.13: Geographic distribution of DenseNet-121 results of ForestsClEvBr class

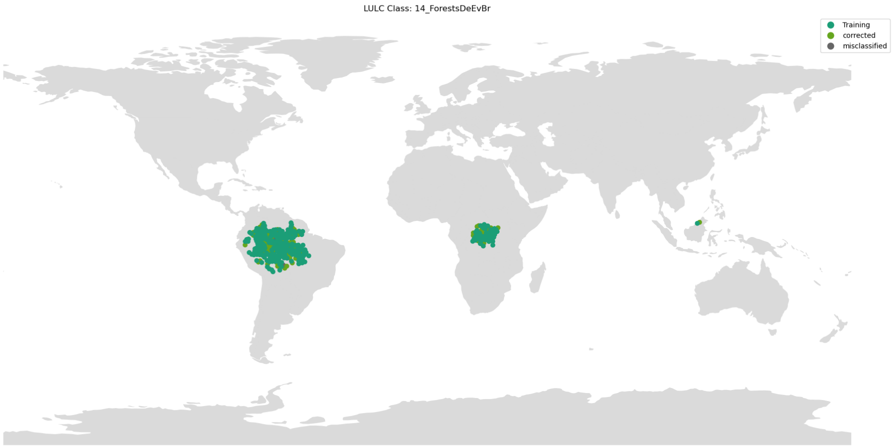


Figure A.14: Geographic distribution of DenseNet-121 results of ForestsDeEvBr class

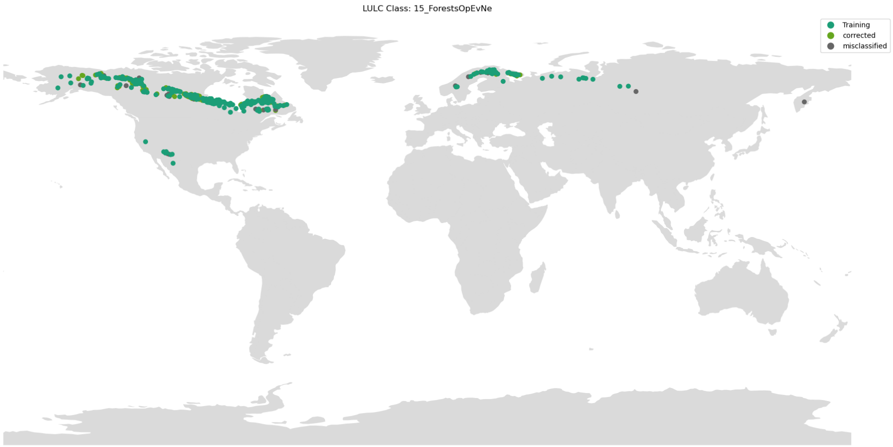


Figure A.15: Geographic distribution of DenseNet-121 results of ForestsOpEvNe class

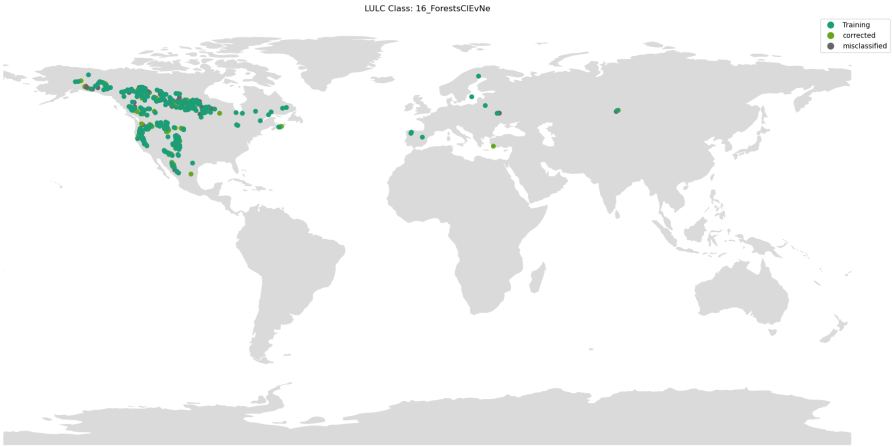


Figure A.16: Geographic distribution of DenseNet-121 results of ForestsClEvNe class

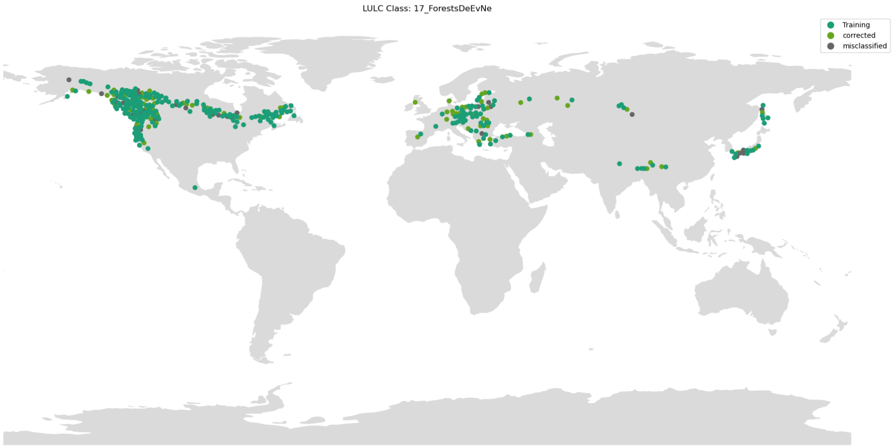


Figure A.17: Geographic distribution of DenseNet-121 results of ForestsDeEvNe class

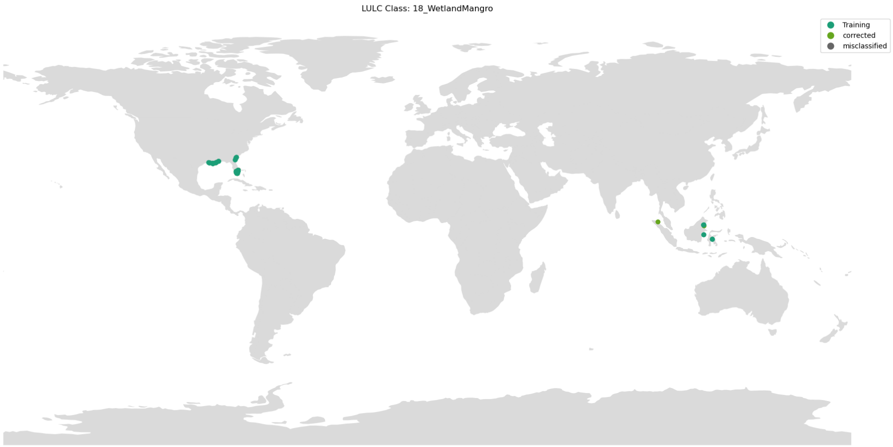


Figure A.18: Geographic distribution of DenseNet-121 results of WetlandMangro class

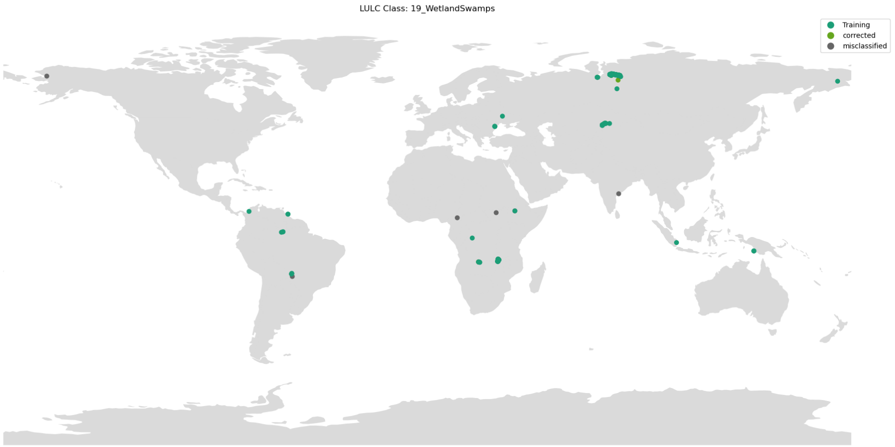


Figure A.19: Geographic distribution of DenseNet-121 results of WetlandSwamps class

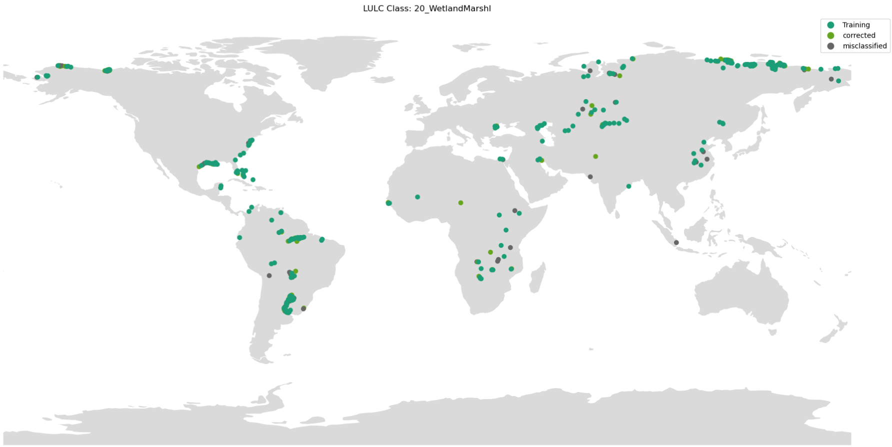


Figure A.20: Geographic distribution of DenseNet-121 results of WetlandMarshl class

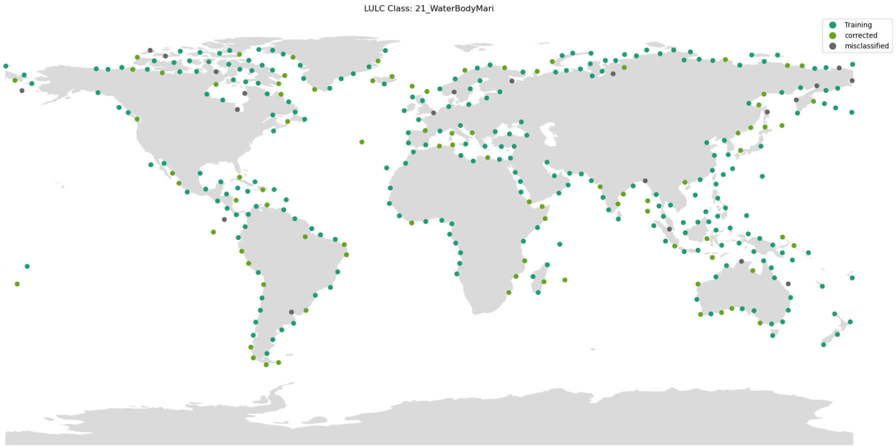


Figure A.21: Geographic distribution of DenseNet-121 results of WaterBodyMari class

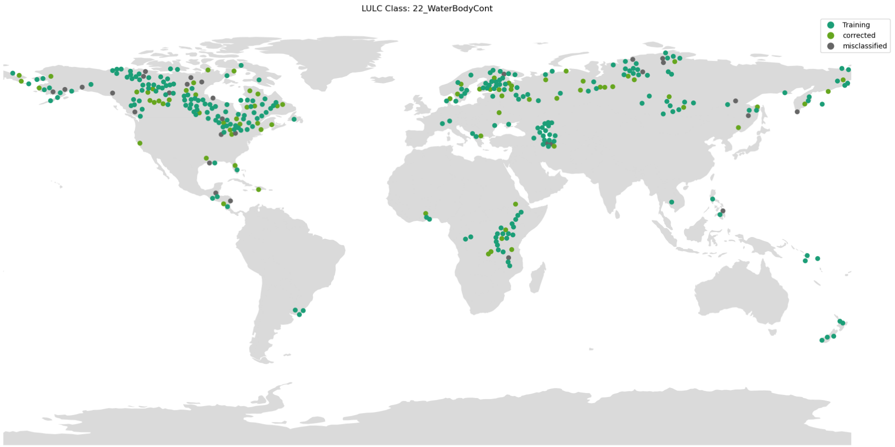


Figure A.22: Geographic distribution of DenseNet-121 results of WaterBodyCont class

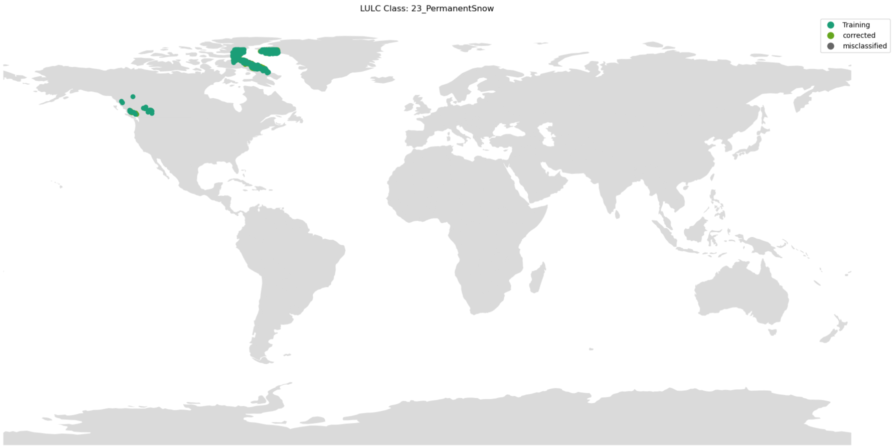


Figure A.23: Geographic distribution of DenseNet-121 results of PermanentSnow class

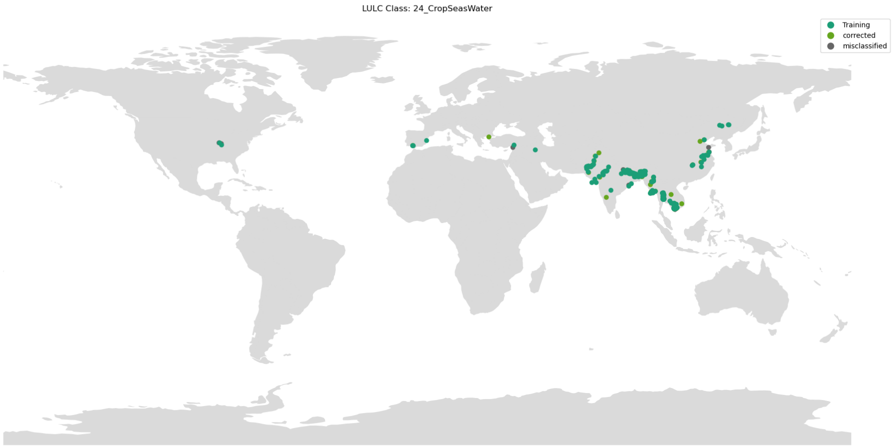


Figure A.24: Geographic distribution of DenseNet-121 results of CropSeasWater class

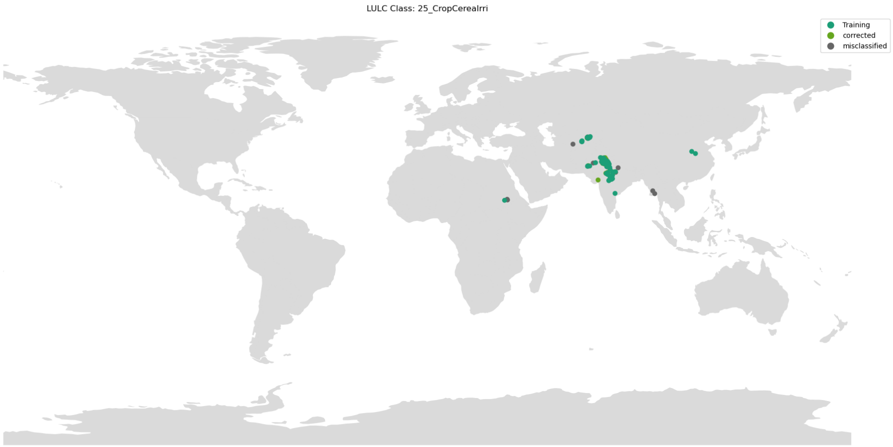


Figure A.25: Geographic distribution of DenseNet-121 results of CropCereaIrri class

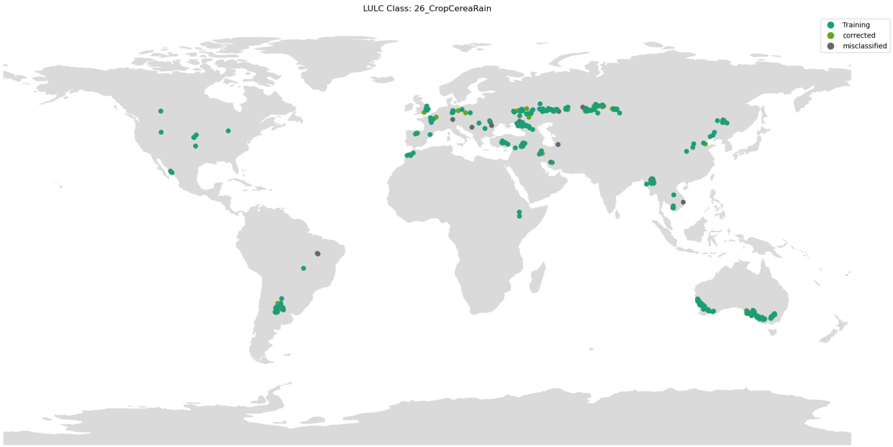


Figure A.26: Geographic distribution of DenseNet-121 results of CropCereaRain class

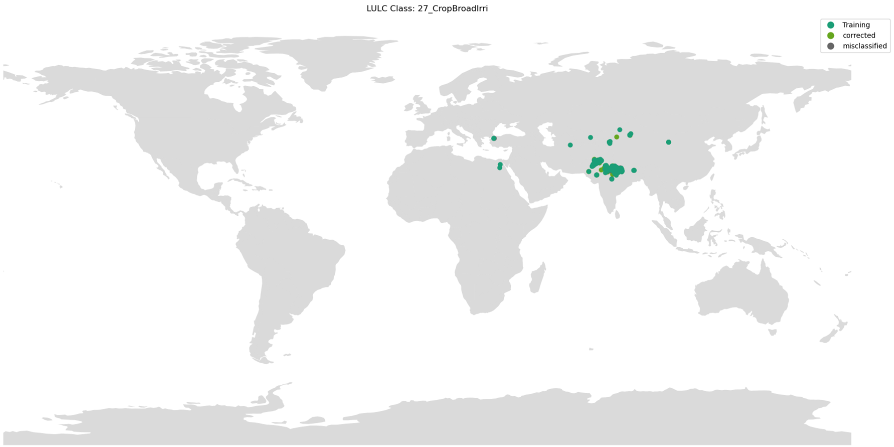


Figure A.27: Geographic distribution of DenseNet-121 results of CropBroadIrri class

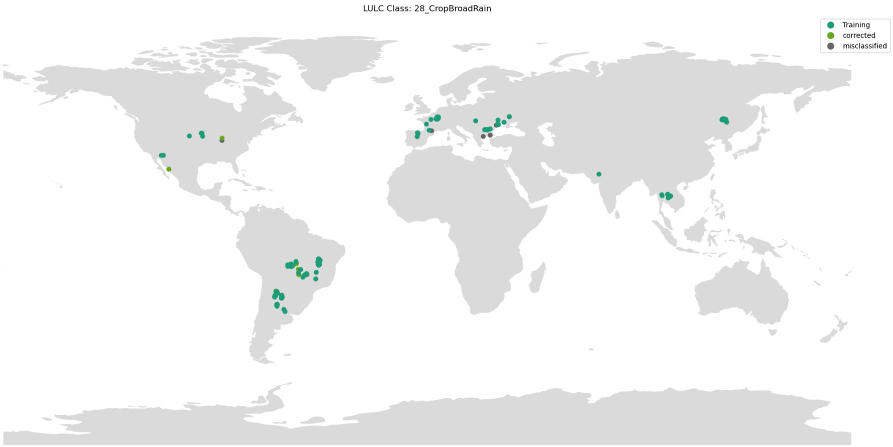


Figure A.28: Geographic distribution of DenseNet-121 results of CropBroadRain class

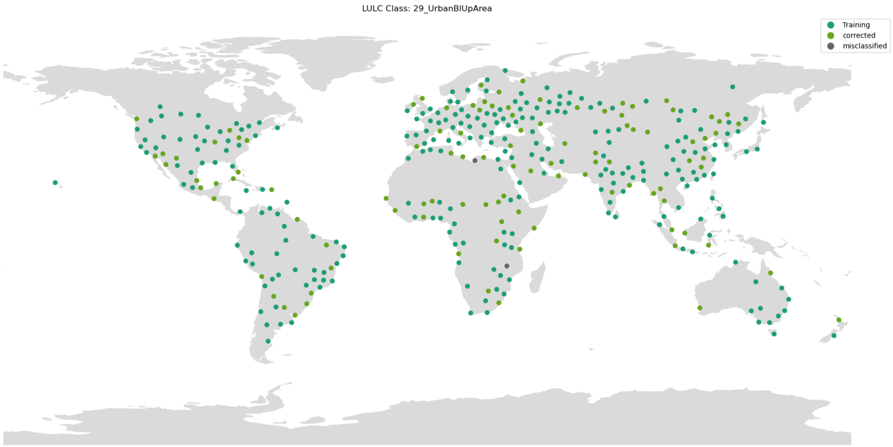


Figure A.29: Geographic distribution of DenseNet-121 results of UrbanBlUpArea class

Bibliography

[1] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for boltzmann machines. Cognitive science, 9(1):147–169, 1985.

[2] S. Akbar, M. Peikari, S. Salama, S. Nofech-Mozes, and A. Martel. Transitioning between convolutional and fully connected layers in neural networks. In Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, pages 143–150. Springer, 2017.

[3] P. F. Alcantarilla, A. Bartoli, and A. J. Davison. Kaze features. In European Conference on Computer Vision, pages 214–227. Springer, 2012.

[4] P. Alirezazadeh, B. Hejrati, A. Monsef-Esfehani, and A. Fathi. Representation learning-based unsupervised domain adaptation for classification of breast cancer histopathology images. Biocybernetics and Biomedical Engineering, 2018.

[5] T. Araújo, G. Aresta, E. Castro, J. Rouco, P. Aguiar, C. Eloy, A. Polónia, and A. Campilho. Classification of breast cancer histology images using convolutional neural networks. PloS one, 12(6):e0177544, 2017.

[6] M. Aswathy and M. Jagannath. Detection of breast cancer on digital histopathology images: present status and future possibilities. Informatics in Medicine Unlocked, 8:74–79, 2017.

[7] J. A. Badejo, E. Adetiba, A. Akinrinmade, and M. B. Akanle. Medical image classification with hand-designed or machine-designed texture descriptors: A performance evaluation. In International Conference on Bioinformatics and Biomedical Engineering, pages 266–275. Springer, 2018.

[8] M. Bakator and D. Radosav. Deep learning and medical diagnosis: A review of literature. Multimodal Technologies and Interaction, 2(3):47, 2018.

[9] D. Bardou, K. Zhang, and S. M. Ahmad. Classification of breast cancer based on histology images using convolutional neural networks. IEEE Access, 6:24680–24693, 2018.

[10] R. Barroso-Sousa and O. Metzger-Filho. Differences between invasive lobular and invasive ductal carcinoma of the breast: results and therapeutic implications. Therapeutic advances in medical oncology, 8(4):261–266, 2016.

[11] E. Bartholome and A. S. Belward. Glc2000: a new approach to global land cover mapping from earth observation data. International Journal of Remote Sensing, 26(9):1959–1977, 2005.

[12] A. N. Basavanhally, S. Ganesan, S. Agner, J. P. Monaco, M. D. Feldman, J. E. Tomaszewski, G. Bhanot, and A. Madabhushi. Computerized image-based detection and grading of lymphocytic infiltration in her2+ breast cancer histopathology. IEEE Transactions on Biomedical Engineering, 57(3):642–653, 2010.

[13] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani. Deepsat: a learning framework for satellite imagery. In Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems, pages 1–10, 2015.

[14] N. Bayramoglu, J. Kannala, and J. Heikkilä. Deep learning for magnification independent breast cancer histopathology image classification. In Pattern Recognition (ICPR), 2016 23rd International Conference on, pages 2440–2445. IEEE, 2016.

[15] Y. Benhammou, B. Achchab, F. Herrera, and S. Tabik. Breakhis based breast cancer automatic diagnosis using deep learning: Taxonomy, survey and insights. Neurocomputing, 375:9–24, 2020.

[16] Y. Benhammou, D. Alcaraz-Segura, E. Guirado, R. Khaldi, B. Achchab, F. Herrera, and S. Tabik. Sentinel2globallulc: A deep-learning-ready sentinel-2 rgb image dataset for global land use/cover mapping. bioRxiv, 2021.

[17] Y. Benhammou, S. Tabik, B. Achchab, and F. Herrera. A first study exploring the performance of the state-of-the art cnn model in the problem of breast cancer. In Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications, pages 1–6, 2018.

[18] A. Bentaieb and G. Hamarneh. Adversarial stain transfer for histopathology image analysis. IEEE transactions on medical imaging, 37(3):792–802, 2018.

[19] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pages 245–250. ACM, 2001.

[20] M. Biswas, V. Kuppili, L. Saba, D. Edla, H. Suri, E. Cuadrado-Godia, J. Laird, R. Marinhoe, J. Sanches, A. Nicolaides, et al. State-of-the-art review on deep learning in medical imaging. Frontiers in bioscience (Landmark edition), 24:392–426, 2019.

[21] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory, pages 144–152. ACM, 1992.

[22] M. Buchhorn, M. Lesiv, N.-E. Tsendbazar, M. Herold, L. Bertels, and B. Smets. Copernicus global land cover layers—collection 2. Remote Sensing, 12(6):1044, 2020.

[23] S. Cascianelli, R. Bello-Cerezo, F. Bianconi, M. L. Fravolini, M. Belal, B. Palumbo, and J. N. Kather. Dimensionality reduction strategies for cnn-based classification of histopathological images. In International Conference on Intelligent Interactive Multimedia Systems and Services, pages 21–30. Springer, 2017.

[24] A. Chan and J. A. Tuszynski. Automatic prediction of tumour malignancy in breast cancer with fractal dimension. Royal Society open science, 3(12):160558, 2016.

[25] J. Chang, J. Yu, T. Han, H.-j. Chang, and E. Park. A method for classifying medical images using transfer learning: A pilot study on histopathology of breast cancer. In e-Health Networking, Applications and Services (Healthcom), 2017 IEEE 19th International Conference on, pages 1–4. IEEE, 2017.

[26] S. Chattoraj and K. Vishwakarma. Classification of histopathological breast cancer images using iterative vmd aided zernike moments & textural signatures. arXiv preprint arXiv:1801.04880, 2018.

[27] D. Chen, S. Liu, P. Kingsbury, S. Sohn, C. B. Storlie, E. B. Habermann, J. M. Naessens, D. W. Larson, and H. Liu. Deep learning and alternative learning strategies for retrospective real-world clinical data. NPJ digital medicine, 2(1):43, 2019.

[28] T. Chen and C. Guestrin. Xgboost: Reliable large-scale tree boosting system. arxiv 2016; 1–6. DOI: http://dx. doi. org/10.1145/2939672.2939785, 2016.

[29] G. Cheng, J. Han, and X. Lu. Remote sensing image scene classification: Benchmark and state of the art. Proceedings of the IEEE, 105(10):1865–1883, 2017.

[30] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and segmentation. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3828–3836. IEEE, 2015.

[31] F. Ciompi, O. Geessink, B. E. Bejnordi, G. S. de Souza, A. Baidoshvili, G. Litjens, B. van Ginneken, I. Nagtegaal, and J. van der Laak. The importance of stain normalization in colorectal tissue classification with convolutional networks. In Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on, pages 160–163. IEEE, 2017.

[32] L. P. Coelho, A. Ahmed, A. Arnold, J. Kangas, A.-S. Sheikh, E. P. Xing, W. W. Cohen, and R. F. Murphy. Structured literature image finder: extracting information from text and images in biomedical literature. In Linking Literature, Information, and Knowledge for Biology, pages 23–32. Springer, 2010.

[33] T. Cohen and M. Welling. Group equivariant convolutional networks. In International conference on machine learning, pages 2990–2999, 2016.

[34] D. Dai and W. Yang. Satellite image classification via two-layer sparse coding with biased image representation. IEEE Geoscience and Remote Sensing Letters, 8(1):173–176, 2010.

[35] K. Das, S. Conjeti, A. G. Roy, J. Chatterjee, and D. Sheet. Multiple instance learning of deep convolutional neural networks for breast histopathology whole slide classification. In Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on, pages 578–581. IEEE, 2018.

[36] K. Das, S. P. K. Karri, A. G. Roy, J. Chatterjee, and D. Sheet. Classifying histopathology whole-slides using fusion of decisions from deep convolutional network on a collection of random multi-views at multi-magnification. In Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on, pages 1024–1027. IEEE, 2017.

[37] R. DeFries. Terrestrial vegetation in the coupled human-earth system: contributions of remote sensing. Annual Review of Environment and Resources, 33:369–390, 2008.

[38] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 248–255. Ieee, 2009.

[39] E. Deniz, A. Şengür, Z. Kadiroğlu, Y. Guo, V. Bajaj, and Ü. Budak. Transfer learning based histopathologic image classification for breast cancer detection. Health information science and systems, 6(1):18, 2018.

[40] A. Di Gregorio. Land cover classification system: classification concepts and user manual: LCCS, volume 2. Food & Agriculture Org., 2005.

[41] M. N. Do and M. Vetterli. The contourlet transform: an efficient directional multiresolution image representation. IEEE Transactions on image processing, 14(12):2091–2106, 2005.

[42] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In International conference on machine learning, pages 647–655, 2014.

[43] K. Dragomiretskiy and D. Zosso. Variational mode decomposition. IEEE transactions on signal processing, 62(3):531–544, 2014.

[44] B. Du, Q. Qi, H. Zheng, Y. Huang, and X. Ding. Breast cancer histopathological image classification via deep active learning and confidence boosting. In International Conference on Artificial Neural Networks, pages 109–116. Springer, 2018.

[45] J. Ferlay, I. Soerjomataram, R. Dikshit, S. Eser, C. Mathers, M. Rebelo, D. M. Parkin, D. Forman, and F. Bray. Cancer incidence and mortality worldwide: sources, methods and major patterns in globocan 2012. International journal of cancer, 136(5):E359–E386, 2015.

[46] J. Ferlay, I. Soerjomataram, M. Ervik, R. Dikshit, S. Eser, C. Mathers, M. Rebelo, D. Parkin, D. Forman, and F. Bray. Globocan 2012 v1. 0, cancer incidence and mortality worldwide: Iarc cancerbase no. 11. lyon, france: International agency for research on cancer; 2013, 2015.

[47] A. Fischer and C. Igel. An introduction to restricted boltzmann machines. In Iberoamerican Congress on Pattern Recognition, pages 14–36. Springer, 2012.

[48] S. Fritz, L. You, A. Bun, L. See, I. McCallum, C. Schill, C. Perger, J. Liu, M. Hansen, and M. Obersteiner. Cropland for sub-saharan africa: A synergistic approach using five land cover data sets. Geophysical Research Letters, 38(4), 2011.

[49] G.-S. Fu, Y. Levin-Schwartz, Q.-H. Lin, and D. Zhang. Machine learning for medical imaging. Journal of healthcare engineering, 2019, 2019.

[50] Z. Gandomkar, P. C. Brennan, and C. Mello-Thoms. Mudern: Multi-category classification of breast histopathological image using deep residual networks. Artificial intelligence in medicine, 2018.

[51] Y. Gao, L. Liu, X. Zhang, X. Chen, J. Mi, and S. Xie. Consistency analysis and accuracy assessment of three global 30-m land-cover products over the european union using the lucas dataset. Remote Sensing, 12(21):3479, 2020.

[52] Z. Gao, L. Wang, L. Zhou, and J. Zhang. Hep-2 cell image classification with deep convolutional neural networks. IEEE journal of biomedical and health informatics, 21(2):416–428, 2017.

[53] E. D. Gelasca, J. Byun, B. Obara, and B. Manjunath. Evaluation and benchmark for biological image segmentation. In Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on, pages 1816–1819. IEEE, 2008.

[54] S. Gengler and P. Bogaert. Combining land cover products using a minimum divergence and a bayesian data fusion approach. International Journal of Geographical Information Science, 32(4):806–826, 2018.

[55] F. Ghaznavi, A. Evans, A. Madabhushi, and M. Feldman. Digital imaging in pathology: whole-slide imaging and beyond. Annual Review of Pathology: Mechanisms of Disease, 8:331–359, 2013.

[56] A. Ghorbanian, M. Kakooei, M. Amani, S. Mahdavi, A. Mohammadzadeh, and M. Hasanlou. Improved land cover map of iran using sentinel imagery within google earth engine and a novel automatic workflow for land cover classification using migrated training samples. ISPRS Journal of Photogrammetry and Remote Sensing, 167:276–288, 2020.

[57] P. Gong, X. Li, J. Wang, Y. Bai, B. Chen, T. Hu, X. Liu, B. Xu, J. Yang, W. Zhang, et al. Annual maps of global artificial impervious area (gaia) between 1985 and 2018. Remote Sensing of Environment, 236:111510, 2020.

[58] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.

[59] N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore. Google earth engine: Planetary-scale geospatial analysis for everyone. Remote sensing of Environment, 202:18–27, 2017.

[60] Z. Guo, L. Zhang, and D. Zhang. A completed modeling of local binary pattern operator for texture classification. IEEE Transactions on Image Processing, 19(6):1657–1663, 2010.

[61] V. Gupta and A. Bhavsar. Breast cancer histopathological image classification: is magnification important? In IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017.

[62] V. Gupta and A. Bhavsar. An integrated multi-scale model for breast cancer histopathological image classification with joint colour-texture features. In International Conference on Computer Analysis of Images and Patterns, pages 354–366. Springer, 2017.

[63] V. Gupta and A. Bhavsar. Sequential modeling of deep features for breast cancer histopathological image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 2254–2261, 2018.

[64] V. Gupta, A. Singh, K. Sharma, and A. Bhavsar. Automated classification for breast cancer histopathology images: Is stain normalization important? In Computer Assisted and Robotic Endoscopy and Clinical Image-Based Procedures, pages 160–169. Springer, 2017.

[65] M. Guray and A. A. Sahin. Benign breast diseases: classification, diagnosis, and management. The oncologist, 11(5):435–449, 2006.

[66] M. N. Gurcan, L. Boucheron, A. Can, A. Madabhushi, N. Rajpoot, and B. Yener. Histopathological image analysis: A review. IEEE reviews in biomedical engineering, 2:147, 2009.

[67] Z. Han, B. Wei, Y. Zheng, Y. Yin, K. Li, and S. Li. Breast cancer multi-classification from histopathological images with structured deep learning model. Scientific reports, 7(1):4172, 2017.

[68] M. C. Hansen, P. V. Potapov, R. Moore, M. Hancher, S. A. Turubanova, A. Tyukavina, D. Thau, S. Stehman, S. J. Goetz, T. R. Loveland, et al. High-resolution global maps of 21st-century forest cover change. science, 342(6160):850–853, 2013.

[69] R. M. Haralick, K. Shanmugam, et al. Textural features for image classification. IEEE Transactions on systems, man, and cybernetics, (6):610–621, 1973.

[70] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[71] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In European conference on computer vision, pages 630–645. Springer, 2016.

[72] P. Helber, B. Bischke, A. Dengel, and D. Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium, pages 204–207. IEEE, 2018.

[73] P. Helber, B. Bischke, A. Dengel, and D. Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 12(7):2217–2226, 2019.

[74] M. Herold, C. E. Woodcock, A. Di Gregorio, P. Mayaux, A. S. Belward, J. Latham, and C. C. Schmullius. A joint initiative for harmonization and validation of land cover datasets. IEEE Transactions on Geoscience and Remote Sensing, 44(7):1719–1727, 2006.

[75] M. H. Hesamian, W. Jia, X. He, and P. Kennedy. Deep learning techniques for medical image segmentation: Achievements and challenges. Journal of digital imaging, pages 1–15, 2019.

[76] G. Hinton, N. Srivastava, and K. Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. Cited on, page 14, 2012.

[77] G. E. Hinton. Deep belief networks. Scholarpedia, 4(5):5947, 2009.

[78] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. science, 313(5786):504–507, 2006.

[79] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, 2012.

[80] R. A. Hoffman, S. Kothari, and M. D. Wang. Comparison of normalization algorithms for cross-batch color segmentation of histopathological images. In Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE, pages 194–197. IEEE, 2014.

[81] Z. Hu, J. Tang, Z. Wang, K. Zhang, L. Zhang, and Q. Sun. Deep learning for image-based cancer detection and diagnosis—a survey. Pattern Recognition, 2018.

[82] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In CVPR, volume 1, page 3, 2017.

[83] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[84] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In Advances in neural information processing systems, pages 2017–2025, 2015.

[85] A. Janowczyk and A. Madabhushi. Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. Journal of pathology informatics, 7, 2016.

[86] N. Japkowicz. The class imbalance problem: Significance and strategies. In Proc. of the Int'l Conf. on Artificial Intelligence, volume 56. Citeseer, 2000.

[87] M. A. Kahya, W. Al-Hayani, and Z. Y. Algamal. Classification of breast cancer histopathology images based on adaptive sparse support vector machine. Journal of Applied Mathematics and Bioinformatics, 7(1):49, 2017.

[88] K. Karra, C. Kontgis, Z. Statman-Weil, J. C. Mazzariello, M. Mathis, and S. P. Brumby. Global land use/land cover with sentinel 2 and deep learning. In 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, pages 4704–4707. IEEE, 2021.

[89] R. Karthiga and K. Narasimhan. Automated diagnosis of breast cancer using wavelet based entropy features. In 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), pages 274–279. IEEE, 2018.

[90] C. M. Kennedy, J. R. Oakleaf, D. M. Theobald, S. Baruch-Mordo, and J. Kiesecker. Managing the middle: A shift in conservation priorities based on the global human modification gradient. Global Change Biology, 25(3):811–826, 2019.

[91] R. Khaldi, D. Alcaraz-Segura, E. Guirado, Y. Benhammou, A. El Afia, F. Herrera, and S. Tabik. Timespec4lulc: A global deep learning-driven dataset of modis terra-aqua multi-spectral time series for lulc mapping and change detection. Earth System Science Data Discussions, 2021:1–28, 2021.

[92] K. Kira and L. A. Rendell. A practical approach to feature selection. In Machine Learning Proceedings 1992, pages 249–256. Elsevier, 1992.

[93] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[94] K. Kumar and A. C. S. Rao. Breast cancer classification of image using convolutional neural network. In 2018 4th International Conference on Recent Advances in Information Technology (RAIT), pages 1–6. IEEE, 2018.

[95] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. nature, 521(7553):436, 2015.

[96] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4):541–551, 1989.

[97] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In Advances in neural information processing systems, pages 396–404, 1990.

[98] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[99] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In Artificial Intelligence and Statistics, pages 562–570, 2015.

[100] S.-J. Lee, T. Chen, L. Yu, and C.-H. Lai. Image classification based on the boost convolutional neural network. IEEE Access, 6:12755–12768, 2018.

[101] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. IEEE transactions on pattern analysis and machine intelligence, 28(9):1465–1479, 2006.

[102] C. Liao, S. Li, and Z. Luo. Gene selection using wilcoxon rank sum test and support vector machine for cancer classification. In International Conference on Computational and Information Science, pages 57–66. Springer, 2006.

[103] M. Lin, Q. Chen, and S. Yan. Network in network. arXiv preprint arXiv:1312.4400, 2013.

[104] L. Liu, X. Zhang, Y. Gao, X. Chen, X. Shuai, and J. Mi. Finer-resolution mapping of global land cover: Recent developments, consistency analysis, and prospects. Journal of Remote Sensing, 2021, 2021.

[105] T. R. Loveland, B. C. Reed, J. F. Brown, D. O. Ohlen, Z. Zhu, L. Yang, and J. W. Merchant. Development of a global land cover characteristics database and igbp discover from 1 km avhrr data. International Journal of Remote Sensing, 21(6-7):1303–1330, 2000.

[106] J. Luengo, D. García-Gil, S. Ramírez-Gallego, S. García, and F. Herrera. Big data preprocessing - enabling smart data. Cham: Springer, 2020.

[107] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson. Deep learning in remote sensing applications: A meta-analysis and review. ISPRS journal of photogrammetry and remote sensing, 152:166–177, 2019.

[108] M. Macenko, M. Niethammer, J. S. Marron, D. Borland, J. T. Woosley, X. Guan, C. Schmitt, and N. E. Thomas. A method for normalizing histology slides for quantitative analysis. In Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on, pages 1107–1110. IEEE, 2009.

[109] A. Maier, C. Syben, T. Lasser, and C. Riess. A gentle introduction to deep learning in medical image processing. Zeitschrift für Medizinische Physik, 29(2):86–101, 2019.

[110] B. B. Mandelbrot. The fractal geometry of nature, volume 1. WH freeman New York, 1982.

[111] D. Marmanis, M. Datcu, T. Esch, and U. Stilla. Deep learning earth observation classification using imagenet pretrained networks. IEEE Geoscience and Remote Sensing Letters, 13(1):105–109, 2015.

[112] J. Martorell-Marugán, S. Tabik, Y. Benhammou, C. del Val, I. Zwir, F. Herrera, and P. Carmona-Sáez. Deep learning in omics data analysis and precision medicine. Exon Publications, pages 37–53, 2019.

[113] M. A. Mazurowski, M. Buda, A. Saha, and M. R. Bashir. Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on mri. Journal of Magnetic Resonance Imaging, 49(4):939–954, 2019.

[114] I. McCallum, M. Obersteiner, S. Nilsson, and A. Shvidenko. A spatial comparison of four satellite derived 1 km global land cover datasets. International Journal of Applied Earth Observation and Geoinformation, 8(4):246–255, 2006.

[115] S. Menke, D. Holway, R. Fisher, and W. Jetz. Characterizing and predicting species distributions across environments and scales: Argentine ant occurrences in the eye of the beholder. Global Ecology and Biogeography, 18(1):50–63, 2009.

[116] K. Michalak and H. Kwasnicka. Correlation based feature selection method. International Journal of Bio-Inspired Computation, 2(5):319–332, 2010.

[117] A. Mikołajczyk and M. Grochowski. Data augmentation for improving deep learning in image classification problem. In 2018 International Interdisciplinary PhD Workshop (IIPhDW), pages 117–122. IEEE, 2018.

[118] J. Morisette, J. Privette, A. Strahler, P. Mayaux, and C. Justice. An approach for the validation of global land cover products through the committee on earth observing satellites, 2003.

[119] N. H. Motlagh, M. Jannesary, H. Aboulkheyr, P. Khosravi, O. Elemento, M. Totonchi, and I. Hajirasouliha. Breast cancer histopathological image classification: A deep learning approach. bioRxiv, page 242818, 2018.

[120] G. Murtaza, L. Shuib, A. W. A. Wahab, G. Mujtaba, H. F. Nweke, M. A. Al-garadi, F. Zulfiqar, G. Raza, and N. A. Azmi. Deep learning-based breast cancer classification through medical imaging modalities: state of the art and research challenges. Artificial Intelligence Review, pages 1–66, 2019.

[121] A.-A. Nahid and Y. Kong. Histopathological breast-image classification using concatenated r–g–b histogram information. Annals of Data Science, pages 1–17, 2018.

[122] A.-A. Nahid and Y. Kong. Histopathological breast-image classification using local and frequency domains by convolutional neural network. Information, 9(1):19, 2018.

[123] A.-A. Nahid, M. A. Mehrabi, and Y. Kong. Histopathological breast cancer image classification by deep neural network techniques guided by local clustering. BioMed research international, 2018, 2018.

[124] A.-A. Nahid, A. Mikaelian, and Y. Kong. Histopathological breast-image classification with restricted boltzmann machine along with backpropagation. Biomedical Research, 29(10):2068–2077, 2018.

[125] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th international conference on machine learning (ICML-10), pages 807–814, 2010.

[126] U. NASS. Usda-national agricultural statistics service, cropland data layer. United States Department of Agriculture, National Agricultural Statistics Service, Marketing and Information Services Office, Washington, DC [Available at http//nassgeodata. gmu. edu/Crop-Scape, Last accessed September 2012.], 2003.

[127] M. A. Nawaz, A. A. Sewissy, and T. H. A. Soliman. Automated classification of breast cancer histology images using deep learning based convolutional neural networks. INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND NETWORK SECURITY, 18(4):152–160, 2018.

[128] E. M. Nejad, L. S. Affendey, R. B. Latip, and I. Bin Ishak. Classification of histopathology images of breast into benign and malignant using a single-layer convolutional neural network. In Proceedings of the International Conference on Imaging, Signal Processing and Communication, pages 50–53. ACM, 2017.

[129] E. M. Nejad, L. S. Affendey, R. B. Latip, I. B. Ishak, and R. Banaeeyan. Transferred semantic scores for scalable retrieval of histopathological breast cancer images. International Journal of Multimedia Information Retrieval, 7(4):241–249, 2018.

[130] A. Ng. Sparse autoencoder, vol. 72 of. CS294A Lecture Notes, 2011.

[131] K. Nogueira, O. A. Penatti, and J. A. Dos Santos. Towards better exploiting convolutional neural networks for remote sensing scene classification. Pattern Recognition, 61:539–556, 2017.

[132] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on pattern analysis and machine intelligence, 24(7):971–987, 2002.

[133] V. Ojansivu and J. Heikkilä. Blur insensitive texture classification using local phase quantization. In International conference on image and signal processing, pages 236–243. Springer, 2008.

[134] J.-F. Pekel, A. Cottam, N. Gorelick, and A. S. Belward. High-resolution mapping of global surface water and its long-term changes. Nature, 540(7633):418–422, 2016.

[135] O. A. Penatti, K. Nogueira, and J. A. Dos Santos. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 44–51, 2015.

[136] L. Perez and J. Wang. The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.04621, 2017.

[137] F. Pérez-Hernández, J. Rodríguez-Ortega, Y. Benhammou, F. Herrera, and S. Tabik. Ci-dataset and detdsci methodology for detecting too small and too large critical infrastructures in satellite images: Airports and electrical substations as case study. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14:12149–12162, 2021.

[138] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In European conference on computer vision, pages 143–156. Springer, 2010.

[139] M. Pfeifer, M. Disney, T. Quaife, and R. Marchant. Terrestrial ecosystems from space: a review of earth observation products for macroecology applications. Global Ecology and Biogeography, 21(6):603–624, 2012.

[140] R. A. Pielke, R. Avissar, M. Raupach, A. J. Dolman, X. Zeng, and A. S. Denning. Interactions between the atmosphere and terrestrial ecosystems: influence on weather and climate. Global change biology, 4(5):461–475, 1998.

[141] S. Pratiher and S. Chattoraj. Manifold learning & stacked sparse autoencoder for robust breast cancer classification from histopathological images. arXiv preprint arXiv:1806.06876, 2018.

[142] T. Quaife, S. Quegan, M. Disney, P. Lewis, M. Lomas, and F. Woodward. Impact of land cover uncertainties on estimates of biospheric carbon fluxes. Global Biogeochemical Cycles, 22(4), 2008.

[143] W. Rawat and Z. Wang. Deep convolutional neural networks for image classification: A comprehensive review. Neural computation, 29(9):2352–2449, 2017.

[144] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. IEEE Computer graphics and applications, 21(5):34–41, 2001.

[145] M. Ringnér. What is principal component analysis? Nature biotechnology, 26(3):303, 2008.

[146] F. Rottensteiner, G. Sohn, J. Jung, M. Gerke, C. Baillard, S. Benitez, and U. Breitkopf. The isprs benchmark on urban object classification and 3d building reconstruction. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3 (2012), Nr. 1, 1(1):293–298, 2012.

[147] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In Computer Vision (ICCV), 2011 IEEE international conference on, pages 2564–2571. IEEE, 2011.

[148] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[149] A. A. Samah, M. F. A. Fauzi, and S. Mansor. Classification of benign and malignant tumors in histopathology images. In Signal and Image Processing Applications (ICSIPA), 2017 IEEE International Conference on, pages 102–106. IEEE, 2017.

[150] D. Sanchez-Morillo, J. González, M. García-Rojo, and J. Ortega. Classification of breast cancer histopathological images using kaze features. In International Conference on Bioinformatics and Biomedical Engineering, pages 276–286. Springer, 2018.

[151] C. Senaras and M. N. Gurcan. Deep learning for medical image analysis. Journal of pathology informatics, 9, 2018.

[152] J. O. Sexton, X.-P. Song, M. Feng, P. Noojipady, A. Anand, C. Huang, D.-H. Kim, K. M. Collins, S. Channan, C. DiMiceli, et al. Global, 30-m resolution continuous fields of tree cover: Landsat-based rescaling of modis vegetation continuous fields with lidar-based estimates of error. International Journal of Digital Earth, 6(5):427–448, 2013.

[153] M. T. Shaban, C. Baur, N. Navab, and S. Albarqouni. Staingan: Stain style transfer for digital histological images. arXiv preprint arXiv:1804.01601, 2018.

[154] M. Sharma, R. Singh, and M. Bhattacharya. Classification of breast tumors as benign and malignant using textural feature descriptor. In Bioinformatics and Biomedicine (BIBM), 2017 IEEE International Conference on, pages 1110–1113. IEEE, 2017.

[155] D. Shen, G. Wu, and H.-I. Suk. Deep learning in medical image analysis. Annual review of biomedical engineering, 19:221–248, 2017.

[156] G. Sheng, W. Yang, T. Xu, and H. Sun. High-resolution satellite scene classification using a sparse coding based multiple feature combination. International journal of remote sensing, 33(8):2395–2412, 2012.

[157] M. J. Shensa. The discrete wavelet transform: wedding the a trous and mallat algorithms. IEEE Transactions on signal processing, 40(10):2464–2482, 1992.

[158] M. Shimada, T. Itoh, T. Motooka, M. Watanabe, T. Shiraishi, R. Thapa, and R. Lucas. New global forest/non-forest maps from alos palsar data (2007–2010). Remote Sensing of environment, 155:13–31, 2014.

[159] A. Shrestha and A. Mahmood. Review of deep learning algorithms and architectures. IEEE Access, 7:53040–53065, 2019.

[160] M. Simard, N. Pinto, J. B. Fisher, and A. Baccini. Mapping forest canopy height globally with spaceborne lidar. Journal of Geophysical Research: Biogeosciences, 116(G4), 2011.

[161] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[162] M. Sokolova, N. Japkowicz, and S. Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In Australasian joint conference on artificial intelligence, pages 1015–1021. Springer, 2006.

[163] Y. Song, H. Chang, Y. Gao, S. Liu, D. Zhang, J. Yao, W. Chrzanowski, and W. Cai. Feature learning with component selective encoding for histopathology image classification. In Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on, pages 257–260. IEEE, 2018.

[164] Y. Song, H. Chang, H. Huang, and W. Cai. Supervised intra-embedding of fisher vectors for histopathology image classification. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 99–106. Springer, 2017.

[165] Y. Song, Q. Li, H. Huang, D. Feng, M. Chen, and W. Cai. Low dimensional representation of fisher vectors for microscopy image classification. IEEE transactions on medical imaging, 36(8):1636–1649, 2017.

[166] Y. Song, J. J. Zou, H. Chang, and W. Cai. Adapting fisher vectors for histopathology image classification. In Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on, pages 600–603. IEEE, 2017.

[167] F. A. Spanhol, L. S. Oliveira, P. R. Cavalin, C. Petitjean, and L. Heutte. Deep features for breast cancer histopathological image classification. In Systems, Man, and Cybernetics (SMC), 2017 IEEE International Conference on, pages 1868–1873. IEEE, 2017.

[168] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte. Breast cancer histopathological image classification using convolutional neural networks. In Neural Networks (IJCNN), 2016 International Joint Conference on, pages 2560–2567. IEEE, 2016.

[169] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte. A dataset for breast cancer histopathological image classification. IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, 63(7):1455, 2016.

[170] F. Statistics. Food and agriculture organization of the united nations. Retrieved, 3(13):2012, 2010.

[171] P. Sudharshan, C. Petitjean, F. Spanhol, L. E. Oliveira, L. Heutte, and P. Honeine. Multiple instance learning for histopathological breast cancer image classification. Expert Systems with Applications, 117:103–111, 2019.

[172] D. Sulla-Menashe and M. A. Friedl. User guide to collection 6 modis land cover (mcd12q1 and mcd12c1) product. USGS: Reston, VA, USA, pages 1–18, 2018.

[173] G. Sumbul, M. Charfuelan, B. Demir, and V. Markl. Bigearthnet: A large-scale benchmark archive for remote sensing image understanding. In IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, pages 5901–5904. IEEE, 2019.

[174] J. Sun and A. Binder. Comparison of deep learning architectures for h&e histopathology images. In Big Data and Analytics (ICBDA), 2017 IEEE Conference on, pages 43–48. IEEE, 2017.

[175] M. Sun, T. X. Han, M.-C. Liu, and A. Khodayari-Rostamabad. Multiple instance learning convolutional neural networks for object recognition. In Pattern Recognition (ICPR), 2016 23rd International Conference on, pages 3270–3275. IEEE, 2016.

[176] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1–9, 2015.

[177] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2818–2826, 2016.

[178] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. IEEE Transactions on Systems, man, and cybernetics, 8(6):460–473, 1978.

[179] P. Teluguntla, P. S. Thenkabail, J. Xiong, M. K. Gumma, C. Giri, C. Milesi, M. Ozdogan, R. Congalton, J. Tilton, T. T. Sankey, et al. Global cropland area database (gcad) derived from remote sensing in support of food security in the twenty-first century: current achievements and future possibilities. 2015.

[180] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. science, 290(5500):2319–2323, 2000.

[181] A. Tharwat. Linear vs. quadratic discriminant analysis classifier: a tutorial. International Journal of Applied Pattern Recognition, 3(2):145–180, 2016.

[182] L. Todorovski and S. Džeroski. Combining classifiers with meta decision trees. Machine learning, 50(3):223–249, 2003.

[183] J. Townshend, C. Justice, W. Li, C. Gurney, and J. McManus. Global land cover classification by remote sensing: present capabilities and future possibilities. Remote Sensing of Environment, 35(2-3):243–255, 1991.

[184] J. R. Townshend and C. O. Justice. Towards operational monitoring of terrestrial systems by moderate-resolution remote sensing. Remote Sensing of Environment, 83(1-2):351–359, 2002.

[185] M.-N. Tuanmu and W. Jetz. A global 1-km consensus land-cover product for biodiversity and ecosystem modelling. Global Ecology and Biogeography, 23(9):1031–1045, 2014.

[186] A. Vali, S. Comai, and M. Matteucci. Deep learning for land use and land cover classification based on hyperspectral and multispectral earth observation data: A review. Remote Sensing, 12(15):2495, 2020.

[187] A. Van Etten, D. Hogan, J. M. Manso, J. Shermeyer, N. Weir, and R. Lewis. The multi-temporal urban development spacenet dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6398–6407, 2021.

[188] B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, and M. Welling. Rotation equivariant cnns for digital pathology. arXiv preprint arXiv:1806.03962, 2018.

[189] R. Venkatesan, P. Chandakkar, and B. Li. Simpler non-parametric methods provide as good or better results to multiple-instance learning. In Proceedings of the IEEE International Conference on Computer Vision, pages 2605–2613, 2015.

[190] P. H. Verburg, K. Neumann, and L. Nol. Challenges in using land use and land cover data for global change studies. Global change biology, 17(2):974–989, 2011.

[191] M. Veta, J. P. Pluim, P. J. Van Diest, and M. A. Viergever. Breast cancer histopathology image analysis: A review. IEEE Transactions on Biomedical Engineering, 61(5):1400–1411, 2014.

[192] C. Wang, J. Shi, Q. Zhang, and S. Ying. Histopathological image classification with bilinear convolutional neural networks. In Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE, pages 4050–4053. IEEE, 2017.

[193] B. Wei, Z. Han, X. He, and Y. Yin. Deep learning model based breast cancer histopathological image classification. In Cloud Computing and Big Data Analysis (ICCCBDA), 2017 IEEE 2nd International Conference on, pages 348–353. IEEE, 2017.

[194] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In Advances in neural information processing systems, pages 1473–1480, 2006.

[195] G. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, and L. Zhang. Aid: A benchmark dataset for performance evaluation of aerial scene classification. arxiv 2016. arXiv preprint arXiv:1608.05167.

[196] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu. Aid: A benchmark data set for performance evaluation of aerial scene classification. IEEE Transactions on Geoscience and Remote Sensing, 55(7):3965–3981, 2017.

[197] G.-S. Xia, W. Yang, J. Delon, Y. Gousseau, H. Sun, and H. Maître. Structural high-resolution satellite image indexing. In ISPRS TC VII Symposium-100 Years ISPRS, volume 38, pages 298–303, 2010.

[198] P. Xu, M. Herold, N.-E. Tsendbazar, and J. G. Clevers. Towards a comprehensive and consistent global aquatic land cover characterization framework addressing multiple user needs. Remote Sensing of Environment, 250:112034, 2020.

[199] L. Yang, S. Jin, P. Danielson, C. Homer, L. Gass, S. M. Bender, A. Case, C. Costello, J. Dewitz, J. Fry, et al. A new generation of the united states national land cover database: Requirements, research priorities, design, and implementation strategies. ISPRS Journal of Photogrammetry and Remote Sensing, 146:108–123, 2018.

[200] Y. Yang and S. Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems, pages 270–279, 2010.

[201] W. Yu, K. Yang, Y. Bai, H. Yao, and Y. Rui. Visualizing and comparing convolutional neural networks. arXiv preprint arXiv:1412.6631, 2014.

[202] N. Zeng, Z. Wang, H. Zhang, W. Liu, and F. E. Alsaadi. Deep belief networks for quantitative analysis of a gold immunochromatographic strip. Cognitive Computation, 8(4):684–692, 2016.

[203] N. Zeng, H. Zhang, B. Song, W. Liu, Y. Li, and A. M. Dobaie. Facial expression recognition via learning deep sparse autoencoders. Neurocomputing, 273:643–649, 2018.

[204] G. Zhang, M. Xiao, and Y.-h. Huang. Histopathological image recognition with domain knowledge based deep features. In International Conference on Intelligent Computing, pages 349–359. Springer, 2018.

[205] L. Zhang, G.-S. Xia, T. Wu, L. Lin, and X. C. Tai. Deep learning for remote sensing image understanding, 2016.

[206] P. Zhang, Y. Ke, Z. Zhang, M. Wang, P. Li, and S. Zhang. Urban land use and land cover classification using novel deep learning models based on high spatial resolution satellite imagery. Sensors, 18(11):3717, 2018.

[207] B. Zhao, Y. Zhong, G.-S. Xia, and L. Zhang. Dirichlet-derived multiple topic scene classification model for high spatial resolution remote sensing imagery. IEEE Transactions on Geoscience and Remote Sensing, 54(4):2108–2123, 2015.

[208] L. Zhao, P. Tang, and L. Huo. Feature significance-based multibag-of-visual-words model for remote sensing image scene classification. Journal of Applied Remote Sensing, 10(3):035004, 2016.

[209] Y. Zheng, Z. Jiang, H. Zhang, F. Xie, Y. Ma, H. Shi, and Y. Zhao. Size-scalable content-based histopathological image retrieval from database that consists of wsis. IEEE journal of biomedical and health informatics, 22(4):1278–1287, 2018.

[210] W. Zhi, H. W. F. Yueng, Z. Chen, S. M. Zandavi, Z. Lu, and Y. Y. Chung. Using transfer learning with convolutional neural networks to diagnose breast cancer from histopathological images. In International Conference on Neural Information Processing, pages 669–676. Springer, 2017.

[211] W. Zhou, S. Newsam, C. Li, and Z. Shao. Patternnet: A benchmark dataset for performance evaluation of remote sensing image retrieval. ISPRS journal of photogrammetry and remote sensing, 145:197–209, 2018.

[212] J. Zhu, S. Rosset, R. Tibshirani, and T. J. Hastie. 1-norm support vector machines. In Advances in neural information processing systems, pages 49–56, 2004.

[213] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint, 2017.

[214] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer. Deep learning in remote sensing: A comprehensive review and list of resources. IEEE Geoscience and Remote Sensing Magazine, 5(4):8–36, 2017.

[215] Q. Zou, L. Ni, T. Zhang, and Q. Wang. Deep learning based feature selection for remote sensing scene classification. IEEE Geoscience and Remote Sensing Letters, 12(11):2321–2325, 2015.