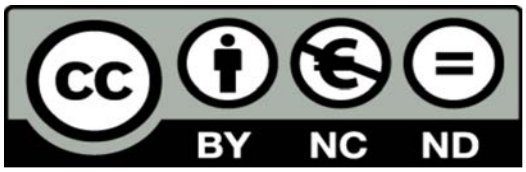
UNIVERSIDAD DE LA RIOJA

TESIS DOCTORAL

Título
Simplifying the usage and construction of deep image classification models
Autor/es
Adrián Inés Armas
Director/es
Jonathan Heras Vicente y Julio Rubio García
Facultad
Facultad de Ciencia y Tecnología
Titulación
Departamento
Matemáticas y Computación
Curso Académico

Tesis presentada como compendio de publicaciones. La edición en abierto de la misma NO incluye las partes afectadas por cesión de derechos



publicaciones.unirioja.es
E-mail: publicaciones@unirioja.es

PhD. thesis

Simplifying the usage and construction of deep image classification models

Adrián Inés Armas

Written under the supervision of
Dr. Jónathan Heras Vicente
Dr. Julio Rubio García

Dissertation submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

Program of Mathematics and Computer Science
Universidad de La Rioja

Logroño, September 2022

Este trabajo ha sido parcialmente subvencionado por una beca FPU 16/06903 del Ministerio de Educación y Ciencia MEC del Gobierno de España, el proyecto MTM2017-88804-P del Ministerio de Economía y Competitividad, el proyecto 2017-I-IDD-00018 de la Agencia de Desarrollo Económico de La Rioja, el proyecto PID2020-115225RB-I00 / AEI / 10.13039/501100011033 del Ministerio de Ciencia e Innovación y las becas ATUR18/21, ATUR19/19, ATUR20/15 de la Universidad de La Rioja.

Agradecimientos

Después de cinco años de un montón de experiencias enriquecedoras y de duro trabajo esta etapa llega a su fin. Durante estos años he vivido una montaña rusa de emociones con algunos de los momentos más felices de mi vida y otros que no lo han sido tanto, y eran en estos momentos difíciles cuando me venía a la mente una de las citas que más me gustan: *"La felicidad puede hallarse hasta en los más oscuros momentos, si somos capaces de usar bien la luz"*. Mi luz en estos años han sido las personas a las que me gustaría dar las gracias en estas líneas.

En primer lugar quiero dar las gracias a todos los miembros del departamento de Matemáticas y Computación. Hace 11 años tuve la suerte de empezar a estudiar en esta universidad. Durante los años de carrera he conocido un montón de profesores que además de transmitirte todo su conocimiento, te transmitían valores y consejos. Cinco años más tarde, Eloy me persuadió para seguir formando parte de este gran grupo y empezar a introducirme en el mundo de la investigación; y finalmente Vico y Jónathan me convencieron para comenzar esta gran aventura. Sin todos ellos y sin todas las enseñanzas recibidas durante mis años en la universidad no estaría hoy aquí escribiendo esto, así que gracias a todos. En especial tengo que dar las gracias a mis directores de tesis, Julio y Jónathan y a mi tutora Vico por guiarme durante mi tesis. Además, tengo que destacar a Jónathan, que ha estado ahí todos los días preocupándose por mi y que más que un director ha sido un apoyo y un amigo. También me gustaría destacar a una persona que he conocido durante mi doctorado, Dani, que se ha convertido en un gran amigo y que con sus tonterías nos alegra los días.

Me gustaría dar las gracias también a mis compañeros de doctorado, Ángela, Jorge, Manu y Patri, sin ellos todo hubiese sido mucho más difícil. He tenido la suerte de compartir esta experiencia con tres amigos de la carrera con los que había compartido comidas, excursiones, fiestas y mucho más. Ahora he podido compartir confesiones y consejos dirección a un congreso, risas y tonterías en los despachos y en los cafés y hemos podido compartir nuestras frustraciones y enfados con los trámites y el papeleo. Tampoco me puedo olvidar de todo lo vivido con Manu, mi compañero de despacho. Como imaginar en aquella cena de fin de curso que nos conocimos que en los siguientes años íbamos a pasar casi todas las horas del día juntos. Hemos podido compartir congresos siendo la pareja de moda, canastas terapéuticas y charlas en las que hemos compartido miedos y frustraciones y también alegrías. Muchas gracias a todos, todo ha sido mucho más fácil gracias a vosotros.

También me gustaría agradecer a todos los amigos que con su compañía este viaje se ha hecho más corto. En especial a la familia Reypa con la que llevo compartiendo

planes toda mi vida y a mi cuadrilla de amigos con los que cada fin de semana me olvidaba de los problemas hablando de cualquier cosa. Tengo que destacar a Omar, que siempre ha estado ahí preocupándose y apoyándome y que gracias a esas largas charlas intentando arreglar el mundo he podido desahogarme y tranquilizarme. También quiero dar las gracias a todas esas personas que han pasado por mi vida y que me han hecho vivir algunos de los momentos más felices de estos años.

Por último, no puedo olvidarme de mi familia, de todos los que están y de los que ya no están. En especial, a mis padres les doy gracias por todo; por esos valores que me han hecho ser quien soy, por el apoyo incondicional pase lo que pase, por esos consejos cuando ni siquiera sabes que los necesitas, por levantarme cuando estoy hundido .... Por todo eso y mucho más, gracias. Os quiero 3000. Y finalmente tengo que nombrar a mi hermana, mi referente. Da igual cuantas veces discutamos y nos enfademos, porque sé que siempre podré contar contigo, sé que siempre estarás ahí para escuchar mis problemas y apoyarme, sé que podré celebrar todas mis alegrías contigo, sé que siempre estarás ahí para guiarme cuando esté perdido. Así que como decían en una de mis películas preferidas que tú y yo veíamos de pequeños: Contigo *"hasta el infinito y más allá"*.

Abstract

Artificial Intelligence, and specifically Deep Learning, has gained great importance in recent years due to the rapid increase in processing capacity, the availability of a large amount of data and the emergence of different open source libraries that allow its use in a simple and free way. Due to this fact, Deep Learning techniques have become the state-of-the-art approach to work on different scientific problems and specifically on image analysis. Image analysis problems often require repetitive time-consuming tasks, and Deep Learning techniques are able to solve these repetitive tasks faster, and in an efficient way. Specifically, these techniques have allowed great advances in different fields such as security, medicine or biology. However, the use of Deep Learning techniques is not trivial since they require a large amount of computational resources, for example specific hardware such as GPUS or TPUS. In addition, it is necessary to have a large amount of annotated data, a requirement that in fields such as medicine or biology can be difficult to fulfil. Finally, expert knowledge of these techniques is required both to build Deep Learning models and to use them. These needs hinder the adoption and democratisation of Deep Learning methods in fields such as medicine or biology where the amount of data resources is limited, and in general, outside the field of computer science due to the need to expert knowledge. Thus, we have identified three challenges related to the use of Deep Learning techniques: the amount of data necessary for the use of these techniques, the democratisation of the construction of Deep Learning models, and the democratisation of the use of Deep Learning models.

The objective of this work is to analyse these challenges and create techniques and tools that help mitigate them in the context of image classification models. First of all, we have focused on reducing the amount of data required to use Deep Learning techniques. In particular, we have developed a framework called CLoDSA that allows anyone to use data augmentation methods for image classification, detection and segmentation problems. In addition, we have created two semi-supervised learning algorithms that allow us to train Deep Learning models using annotated and non-annotated data. The first algorithm is based on data and model distillation, whereas the second uses topological data analysis techniques.

In order to democratise the construction of Deep image classifications models, we have developed an AutoML tool, called ATLASS, which assists the user in the entire process of creating an image classification model, from annotating the images, yo the creation and usage of such a Deep Learning model. This tool has been validated with several datasets obtaining better results than other AutoML tools.

The problem of democratising the use of Deep Learning models has been approached in two different ways. In the first place, to reduce the amount of resources needed to use Deep Learning models, we have studied the combination of a semi-supervised method with compact networks and quantification techniques. This approach has reduced the amount of computational resources needed to train and use Deep Learning models. Moreover, models created with this approach have similar or even better performance than standard size models and are also faster and lighter. Secondly, the democratisation of the use of Deep Learning models has been addressed by creating a framework called DeepClas4Bio, which provides a common access point for the classification models of various Deep Learning libraries and facilitates the interoperability of bioimaging tools with Deep Learning models. In addition, a series of plugins have been created to connect the main biomedical tools with such a framework.

Finally, the aforementioned techniques have been the basis to deal with two real biomedical problems: the measurement of the propagation of bacteria in motility images, and the detection of diseases of the epiretinal membrane from fundus images.

Resumen

La Inteligencia Artificial, y en concreto el Aprendizaje Profundo (en inglés *Deep Learning*), ha cobrado gran importancia en los últimos años debido al rápido aumento de la capacidad de procesamiento, a la disponibilidad de una gran cantidad de datos y al surgimiento de diferentes librerías de código abierto que permiten su uso de manera sencilla y libre. Es por esto que las técnicas de *Deep Learning* se han convertido en el estado del arte para trabajar en diferentes problemas científicos y en concreto en el análisis de imágenes. A menudo, los problemas de análisis de imágenes requieren realizar tareas repetitivas que consumen una gran cantidad de tiempo, y las técnicas de *Deep Learning* son capaces de resolver estas tareas repetitivas de manera más rápida y eficiente. En concreto, estas técnicas han permitido grandes avances en diferentes campos como la seguridad, la medicina o la biología. Sin embargo, el uso de las técnicas de *Deep Learning* no es trivial. Para su uso es necesario contar con una gran cantidad de recursos computacionales, por ejemplo hardware especifico como GPUS o TPUS. Además, es necesario contar con una gran cantidad de datos anotados, algo que en campos como la medicina o la biología puede ser difícil de conseguir. Por último, es necesario contar con un conocimiento experto de estas técnicas tanto para construir modelos de *Deep Learning* como para usarlos. Estas necesidades dificultan la adopción y democratización del *Deep Learning* en campos como la medicina o la biología, donde la cantidad de recursos de datos es limitada, y en general, en cualquier campo distinto a las ciencias de la computación debido a la necesidad de un conocimento experto. Así, hemos identificado tres desafios relacionados con el uso de las técnicas de *Deep Learning*, que son: la cantidad de datos necesaria para el uso de estas técnicas, la democratización de la construcción de modelos de *Deep Learning* y la democratización del uso de modelos de *Deep Learning*.

El objetivo de este trabajo es analizar estos desafios y crear técnicas y herramientas que ayuden a mitigarlos en el contexto de la clasificación de imágenes. En primer lugar nos hemos centrado en reducir la cantidad de datos necesarios para usar las técnicas de *Deep Learning*. Para ello, hemos desarrollado un *framework* llamado CLoDSA que permite realizar aumento de datos para problemas de clasificación, detección y segmentación de imágenes. Además, hemos creado dos algoritmos de aprendizaje semisupevisado que permiten entrenar modelos de *Deep Learning* usando datos anotados y sin anotar. El primer algoritmo está basado en la destilación de datos y modelos, mientras que el segundo utiliza técnicas del análisis topológico de datos.

Con el fin de democratizar la construcción de modelos de *Deep Learning*, hemos desarrollado una herramienta de *AutoML*, llamada ATLASS, que asiste al usuario en

todo el proceso de creación de un modelo de clasificación de imágenes, desde la anotación de las imágenes, hasta la creación y uso de dicho modelo de *Deep Learning*. Esta herramienta ha sido validada con varios datasets obteniendo mejores resultados que otras herramientas de *AutoML*.

El problema de la democratización del uso de modelos de *Deep Learning* se ha abordado de dos maneras distintas. En primer lugar, para tratar de reducir la cantidad de recursos necesarios para el uso de estos modelos, se ha estudiado la combinación de métodos semi-supervisados con redes compactas y técnicas de quantificación, lo que ha permitido reducir la cantidad de recursos computacionales necesarios para entrenar y usar modelos de *Deep Learning*. Los modelos creados con esta aproximación tienen un rendimiento similar, o incluso superior, a los modelos de tamaño estandar y además son más rápidos y ligeros. En segundo lugar se ha abordado la democratización del uso de modelos de *Deep Learning* creando un *framework* llamado DeepClas4Bio, que proporciona un punto de acceso común para los modelos de clasificación de varias librerías de *Deep Learning* y facilita la interoperabilidad de las herramientas de bioimagen con modelos de *Deep Learning*. Además, se han creado una serie de *plugins* para la conexión de las principales herramientas biomédicas con dicho *framework*.

Por último, las técnicas y herramientas nombradas previamente han sido la base para abordar dos problemas biomédicos reales, como son la medición de la propagación de bacterias en imágenes de motilidad y la detección de enfermedades de la membrana epirretiniana a partir de imágenes de fondo de ojo.

Contents

Publications **1**

1 Introduction **3**
1.1 Motivation 3
1.2 Challenge 1: The amount of resources 6
1.3 Challenge 2: Democratisation of models' construction 10
1.4 Challenge 3: Democratisation of models' usage 12
1.5 Objectives 14

2 Results and discussion **15**
2.1 Objective O1: Reducing the amount of data resources 15
2.2 Objective O2: Democratising Deep Learning models' construction . . 26
2.3 Objective O3: Reducing the amount of computational resources . . . 27
2.4 Objective O4: Democratising Deep Learning models' usage 30
2.5 Objective O5: Applying developed methods 32

3 Contributions **39**
3.1 DeepClas4Bio: Connecting bioimaging tools with Deep Learning frameworks for image classification 39
3.2 CLoDSA: A tool for augmentation in classification, localization, detection, semantic segmentation and instance segmentation tasks 41
3.3 Biomedical image classification made easier thanks to transfer and semi-supervised learning 43
3.4 MotilityJ: An open-source tool for the classification and segmentation of bacteria on motility images 45
3.5 Prediction of Epiretinal Membrane from Retinal Fundus Images Using Deep Learning 47
3.6 Semi-Supervised Learning for Image Classification using Compact Networks in the BioMedical Context 48

4 Conclusions and further work **51**

5 Conclusiones y trabajo futuro **53**

Publications

This memoir is organised as a compendium of the following research papers:

- A. Inés, C. Domínguez, J. Heras, E. Mata, and V. Pascual. "DeepClas4Bio: Connecting bioimaging tools with Deep Learning frameworks for image classification". In: Computers in Biology and Medicine 108 (2019), pp. 49–56. ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2019.03.026. Impact Factor: 3.434, Q1(JCR).
- A. Inés, C. Domínguez, J. Heras, E. Mata, and V. Pascual. "Biomedical image classification made easier thanks to transfer and semi-supervised learning". In: Computer Methods and Programs in Biomedicine 198 (2021), p. 105782. ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2020.105782. Impact Factor: 5.428, Q1(JCR).
- A. Casado-García, G. Chichón, C. Domínguez, M. García-Domínguez, J. Heras and A. Inés, M. López, E. Mata, V. Pascual, and Y. Sáenz. "MotilityJ: An open-source tool for the classification and segmentation of bacteria on motility images". In: Computers in Biology and Medicine 136 (2021), p. 104673. ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2021.104673. Impact Factor: 4.589, Q1(JCR).

The rest of the memoir is organised as follows. The first chapter presents the motivation of our work, the problems that have been addressed and the objectives that we would like to achieve. The second chapter shows the results obtained with our work, and Chapter 3 summarises our contributions. Finally, in the last chapter, we present the conclusions of this work and future lines of research. As appendixes, we provide the research papers that form this work.

All the code associated with the projects developed in this work can be found in the following repository https://github.com/adines/Thesis.

Chapter 1

Introduction

In this chapter, we present the general context wherein our work is framed. In addition, we study the challenges and problems that arise within the field of study of this work, explaining in detail how they have been addressed in the literature, and the limitations of those approaches. Finally, we set our objectives.

1.1 Motivation

Artificial Intelligence is a term that was coined in 1956 at the Dartmouth College Artificial Intelligence Conference [1], but it has been in recent years when it has become one of the most important and promising fields of research in Computer Science. Artificial Intelligence is a set of techniques that try to imitate the capacity for analysis and decision-making of human beings, allowing computers to solve repetitive problems and labour-intensive human tasks, but in a fast and automatic way [2]. Thus, the advancement of this technology not only brings benefits and results in the field of Computer Science, but it is also used in fields as diverse as Robotics [3], Biology [4] or Medicine [5], among others. However, despite the fact that the usage of Artificial Intelligence is widespread, and that we often interact with this technology in our daily lives, the adoption of these new techniques is not straightforward in many fields. This is because most users do not know how Artificial Intelligence works, and it is simply seen as a black box that magically performs tasks [6]. Therefore, for Artificial Intelligence researchers, in addition to make new advances in the field, one of their most important tasks should be to democratise these new techniques and develop human-centred systems which bring Artificial Intelligence closer to the real needs of people [7].

Within Artificial Intelligence, Computer Vision is a discipline where the democratisation of the tools is utterly important. Computer Vision methods are focused on analysing and processing images and videos, and extracting information from them. Some examples where Computer Vision methods are applied are: disease detection [8], facial recognition [9], or autonomous driving [10]. These applications are usually based on classifying, detecting, measuring or counting elements within an image or video; and most of these tasks might seem simple since human beings carry them out

instinctively, but we have been trained throughout our entire lives to conduct them in an automatic way. However, these tasks are complicated for a computer due to the *semantic gap* [11]. This term refers to the fact that computers see images as matrices, which entails a series of challenges such as changes in point of view or scale, lighting changes, or warping, that make studying images with a computer complex. In addition, as the number of images increases, manually analysing images becomes an unaffordable task; and, therefore, computer-based methods are required. A context where this is common is life-science research.

The analysis of images in life-science is, in many cases, conducted manually [12]. This is a tedious, expensive and unreproducible process; specially, when the number of images increases — a common scenario due to the new image acquisition methods [13]. Therefore, computer based methods are gaining great importance to tackle life-science tasks such as the diagnosis of diseases not only in humans [8] but also in plants [14], drug discovery [15], or monitoring climate change [16]. These tasks involve solving problems such as identifying, detecting and segmenting objects of interest in a given image; and one of the most common problems in this context is image classification.

Image classification is a computer vision task that assigns a label chosen from a fixed set of categories to a given image. Image classification methods have become an instrumental tool in life-sciences for tasks such as the classification of breast cancer histology images [5], the classification of echocardiograms [17], the detection and classification of nuclei in routine colon cancer histology images [18], the classification of lung nodules on computed tomography images [19], or the classification of skin cancer images [20]. Image classification problems have been tackled in the literature using different techniques [21–23], but Machine Learning and more concretely *Deep Learning* methods have offered the best results in recent years.

Up to 2010, the traditional approach to deal with image classification tasks was to use supervised Machine Learning techniques. Machine Learning is a set of Artificial Intelligence techniques based on the idea that systems can learn from data, identify patterns, and make decisions with minimal human intervention [24]. Specifically, supervised Machine Learning techniques are based on optimising a set of parameters from a mathematical function using annotated data (that is, data with the expected response value) [25]. Such a process is known as *training* and produces a model (the optimised function) that allows us to extract information for new data. In the context of image classification, the classification process had two phases, see Figure 1.1. In a first step, a series of descriptors were extracted from the image, using techniques such as LBP [26], SIFT [27], SURF [28], or ORB [29]. These descriptors were the input data of the Machine Learning model, called the classifier. In a second step, this model was trained with these descriptors. Finally, the trained model allowed us, from the descriptors of a new image, to classify it. The main problem with this process is that its performance is closely linked to the selection of good descriptors. This fact makes expert knowledge necessary for their extraction. Furthermore, this process is specific to each problem, which makes automation more difficult and avoids reusing the knowledge learned in one task for another. Deep Learning methods allow us to solve these problems because they learn the necessary descriptors for a given problem automatically.

The explosion of Deep Learning techniques has been one of the fundamental rea-

son of the rise of Computer Vision and Artificial Intelligence in the last 10 years [30]. Deep Learning is a set of Machine Learning techniques, see Figure 1.2, that are based on neural networks, and that arose from the idea of the perceptron developed by Frank Rosenblatt in 1959 [31]. These networks offer numerous layers of abstraction that make possible to automatically learn representations of unstructured data such as images, text or audio. In particular, the layers of these networks allow us to learn the descriptors of the data, from very basic descriptors in the lower layers to high-level descriptors in the upper layers. There are several types of neural networks including Convolutional Neural Networks (CNNs) [32], Recurrent Neural Networks [33], Graph Neural Networks [34] or Transformers [35]; and CNNs, and recently Transformers, are the most widely used networks for Computer Vision since lines, edges or shapes are detected in the lower layers, and as we progress through the layers, more complex and specific features related to the particular problem are learned, such as eyes, ears or even faces [36]. In this way, the learning process of Deep Learning models is carried out in a single step where the descriptors are learned and the model is trained, unlike the traditional approach where the learning process was carried out in two steps, see Figure 1.1.

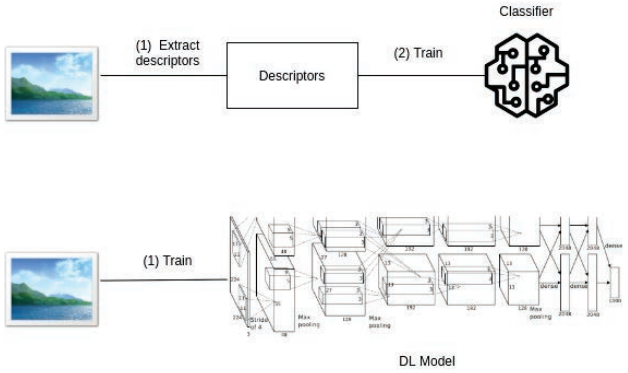


Figure 1.1: Comparative diagram of the traditional approach versus the Deep Learning approach.

Deep Learning techniques have gained importance in recent years due to three reasons [37]. First of all, the fast increase in computer processing capacity, with the creation of specific hardware such as GPUs (Graphical Processing Units) or TPUs (Tensor Processing Units) which allows us to train Deep Learning models in a reasonable time. Secondly, the availability of a large amount of data, an instrumental ingredient to train Deep Learning models, has considerably grown. And finally, the availability of open-source libraries that allow scientists to use these techniques simply and freely. For all these reasons, Deep Learning techniques have become part of the toolbox of numerous life scientists, allowing numerous advances in the analysis of images, texts or audios, and facilitating numerous contributions in different fields [13]. However, the use of these techniques is not straightforward and they have different limitations and problems that must be addressed. In this work, we have focused on three of them that are especially relevant in the context of image classification in life-sciences: the

great amount of resources (both data and computational) that are required to train the Deep Learning models, the challenges of constructing Deep Learning models, and, the difficulties of using trained Deep Learning models.

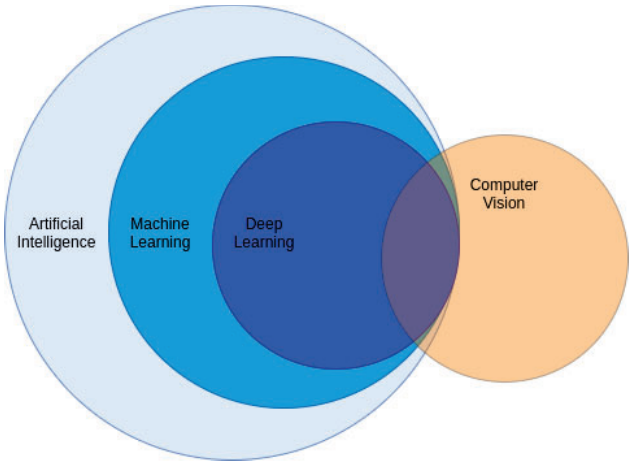


Figure 1.2: General diagram of the relations among Artificial Intelligence, Machine Learning, Deep Learning and Computer Vision.

The next three sections are devoted to explain these problems in detail and to present the available solutions in the literature.

1.2 Challenge 1: The amount of resources

Deep Learning methods are data hungry; that is, they require a great amount of annotated images to obtain good results [38]. At first glance, this may not seem a problem due to the large amount of images that is generated in a daily basis, and the existence of large datasets such as Imagenet [39], with more than 14 million images; CoCo [40], with more than 330 thousand images; or JFT [38], with 300 million images. However, image acquisition is not easy in some biomedical contexts due to, for example, a limited budget to obtain samples, the privacy of the data, the need to perform an invasive medical procedure, or destructive imaging processes that limit the amount of images that can be acquired. In addition, the images have to be annotated. This means that, for instance, in the image classification context for each image we must provide its associated label. The annotation of images is one of the main problems of Deep Learning because it is a very time-consuming task and, in biomedical contexts, it requires expert knowledge. For this reason, the application of Deep Learning models in the biomedical field is not straightforward [41] and it is necessary to use different techniques, such as transfer learning, semi-supervised learning or data augmentation, that allow us to apply Deep Learning techniques without having a large amount of annotated data. In addition, Deep Learning models are not only data demanding, but also, computationally demanding in training time. For instance, training ResNet-50, an image classification

model, with the ImageNet dataset took 8.7 minutes in 1024 Pascal GPUs [42], and such an amount of computational resources is not affordable by everyone.

Transfer learning is one of the most used techniques to solve the problem of working with limited resources [43]. This technique has been successfully used for classifying medical images [44], high-resolution images [45] and brain tumours in MR images [46]. Transfer learning consists in learning a general representation for solving a problem where we have enough annotated data, and, then, transferring such a knowledge to a more specific task wherein we have less data. As we can see in Figure 1.3, in the first stage of transfer learning, we (1) train a model in a general task of classifying natural images using a dataset like ImageNet [39], which allows the models to learn a large number of useful features. Then, in a second stage, we (2) initialise a model with the weights learned in (1), and (3) refine this model using a small dataset for a particular task. The problem with this approach is that when the domain of the target task is far from the domain of the source task; that is, the characteristics of the target images differ considerably from the characteristics of the source images, transfer learning does not obtain results as good as when the domains of the tasks are closer [47, 48]. This happens especially when working with biomedical images such as magnetic resonance images, OCT images or infrared images. Namely, the datasets available in the biomedical field are usually relatively small compared to the large datasets like ImageNet, CoCo or JFT used for transfer learning, and when models are trained in those small biomedical datasets, problems such as *overfitting* arise [49].

Deep Learning networks are giant mathematical models capable of learning a large number of features; however, when the number of examples used for learning is not large enough, these models are capable of memorising all the characteristics of these examples, resulting in a model that perfectly recognises the training data but it is unable to generalise this knowledge to new samples. This problem is called overfitting, see Figure 1.4. In order to avoid overfitting, regularisation techniques [50] can be applied during the training process. The main regularisation techniques used in the literature are L1 and L2 regularisation [51], dropout [52], data augmentation [53] and early stopping [54]. Among them, the technique that offers the best results is data augmentation [53]. Data augmentation, see Figure 1.5, consists in increasing the number of training images by carrying out small transformations that do not alter the class of the images, and then training a model with both the initial images and the augmented ones. Some of those transformations are translations, rotations or colour changes. The problem with these transformations is that they cannot always be applied since sometimes they can change the interpretation of the image when applied to them. In the classification context, transformation techniques for image augmentation do not generally change the class of an image, but they might alter the annotation in other problems like detection or segmentation. For instance, applying the vertical flip operation to a melanoma image does not change the class of the image; but the position of the melanoma in the new image has changed from the original image. Therefore, it is necessary to develop different techniques for increasing data suitable for each specific task.

Data augmentation makes possible to largely fix the overfitting problem produced when we have few data points by introducing new examples artificially. However, in many cases this increment is insufficient and it is necessary to obtain more annotated

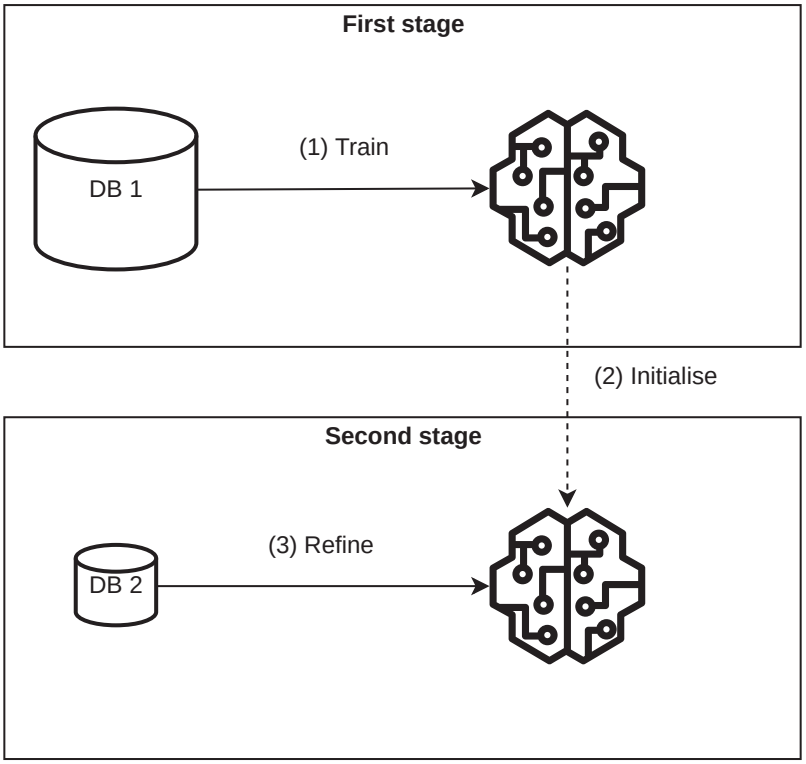


Figure 1.3: General diagram of the training process of a Deep Learning model using transfer learning.

data. Sometimes, it is relatively easy to acquire new images but label them becomes a complicated task. To solve this problem, two learning methods have arisen: semi-supervised [55] and self-supervised [56] methods.

Semi-supervised learning methods use both labelled and unlabelled data, whereas self-supervised methods use only unlabelled data. In particular, as we can see in Figure 1.6, semi-supervised methods are an approach that in general (1) defines a base model that is trained on labelled data, (2) uses the model to predict labels for unlabelled data, and, finally, (3) initialise a model with the weights learned in (1), and (4) retrains the model with both the most confident predictions produced in (2) and the initial data; thus, enlarging the labelled training set. Semi-supervised learning methods can be grouped into three main types: self-training, consistency regularisation and hybrid methods [57]. In self-training methods, a model is trained on labelled data and used to predict pseudo-labels for the unlabelled data. The model is then trained on both ground truth labels and pseudo-labels simultaneously. Some examples of these methods are pseudo label [58] and noisy student [59]. Consistency regularisation methods, such as virtual adversarial training [60], mean teacher [61] or π-models [62]; use the idea that the model prediction on an unlabelled image should remain the same even

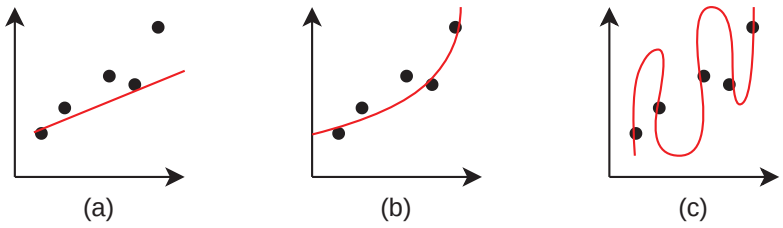


Figure 1.4: (a) Underfitting, (b) correct desired performance and (c) overfitting.

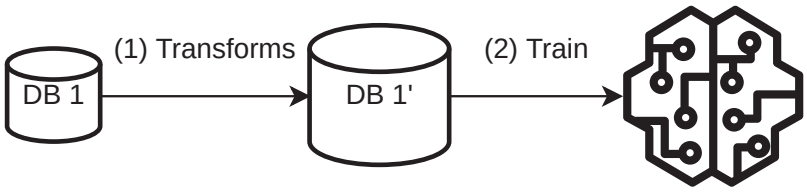


Figure 1.5: General diagram of the training process of a Deep Learning model using data augmentation.

after adding some noise. Finally, hybrid methods combine ideas from self-training and consistency regularisation along with additional components for performance improvement. These methods include FixMatch [63] and MixMatch [64].

In the case of self-supervised methods, see Figure 1.7, they (1) define a base model that is trained on unlabelled images using an auxiliary task that aims to learn a general representation of images. Then, (2) a model is initialised with the weights learned in (1) and retrained with the labelled data. Self-supervised methods learn useful representations without using a large amount of labelled data. These auxiliary tasks employed in self-supervised learning can be grouped into three main types: image reconstruction [65], visual common sense tasks [66] and contrastive learning [67]. Image reconstruction refers to tasks that consist in correcting the information of an image, which is either corrupt or has partial only information. Examples of these tasks are denoising autoencoders [68], in-painting [69] or colourisation [69]. Visual common sense tasks, such as rotation [70], relative position [66] or jigsaw puzzles [66], focus on the positions of the main features within an image by detecting possible changes such as rotations or rearrangements. Finally, contrastive learning methods [67] are a popular form of self-supervised learning that aim to learn representations by enforcing similar elements to be equal, and dissimilar elements to be different. The main problem with all self-supervised learning methods is that they have been mainly tested with huge natural image datasets [56], and it is not clear whether they are applicable to other kind of images, in particular in life-science.

In this section, we have seen that to solve the problem of the large amount of annotated images necessary to obtain a good performance of Deep Learning techniques, there are several alternatives, such as transfer learning, data augmentation or semi-

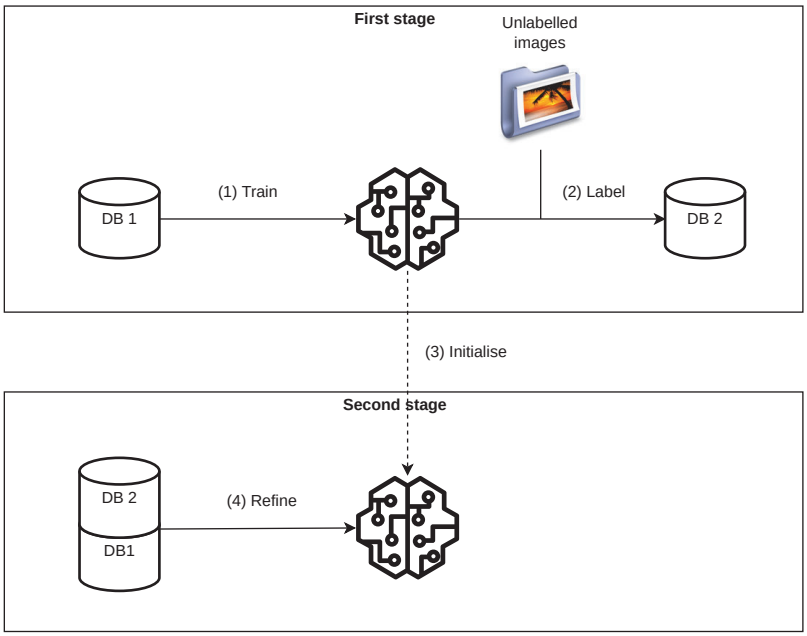


Figure 1.6: General diagram of the training process of a Deep Learning model using semi-supervised learning.

supervised and self-supervised techniques, but they still have some limitations. In addition, except in the case of data augmentation, most of these techniques are not implemented in simple to use libraries that facilitate their use. Due to this fact, it is difficult to create Deep Learning models using these techniques as we will see in the next section.

1.3 Challenge 2: Democratisation of models' construction

The construction of Deep Learning models is not straightforward since it often requires expert knowledge about libraries and tools that do not usually belong to the toolbox of life scientists. For example, some programming and image processing skills and some experience with Deep Learning techniques are required. In addition, there are currently a large number of approaches that allow users to solve the same problem (for instance, the timm library[1] provides more than 300 models for image classification); so, it is interesting to try different alternatives [71]. This can be tricky due to the large number of different frameworks, models, settings and hyperparameters of each alternative. This selection process is usually called a Machine Learning pipeline. Unfortunately, there

[1] `https://timm.fast.ai/`

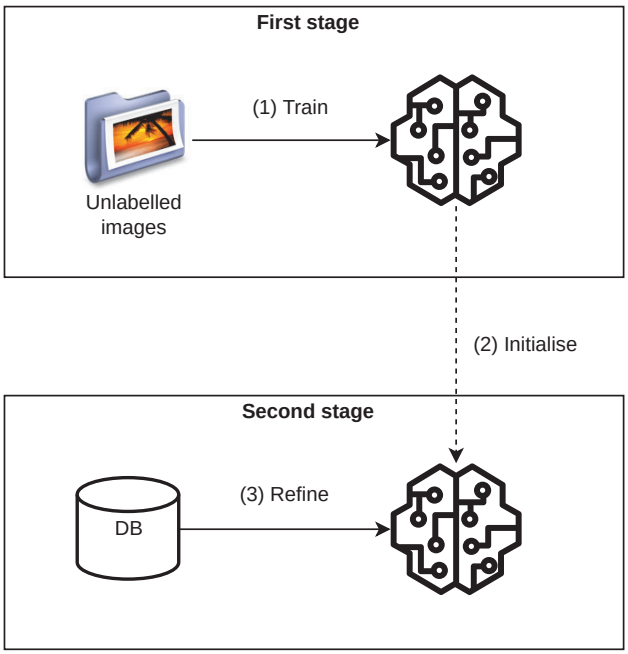


Figure 1.7: General diagram of the training process of a Deep Learning model using self-supervised learning.

is no rule of thumb in about which direction to go to select the correct parameters in such a pipeline.

To solve this problem, AutoML systems have arisen [72]. These systems aim to facilitate the construction of models for domain experts with a limited Machine Learning background. Since there are a lot of alternatives for each step of a Machine Learning pipeline (for instance, the algorithm used, or the selection of hyperparameters) AutoML techniques try to find the best combination for each particular problem. In the context of image classification, we can find different AutoML tools, such as, Google AutoML [73], AutoKeras [74] or AutoGluon [75]. However, these systems are focused on Neural Architecture Search (NAS) methods [76] for image recognition, a technique for automating the design of artificial neural networks. NAS methods learn a network topology that can achieve the best performance on a certain task based on three components: search space, search algorithm and child model evolution strategy. Even if this approach has outperformed manually designed architectures [77], the adoption of NAS techniques by non-expert users to construct recognition models is far from trivial; mainly, because these techniques are computationally intensive (for instance, the design of the NasNet architecture took 1800 GPU days [77], and the AmoebaNet architecture took 3150 GPU days [78]) and require huge datasets (usually, datasets like ImageNet [39] or CIFAR [79], that contain hundred of thousands, or even millions, annotated images). In addition, NAS techniques require some prior knowledge about

typical properties of Deep Learning architectures to simplify the search [76], and also require experience to configure the tools implementing those methods.

Another limitation of AutoML tools is that they do not include techniques such as transfer learning, data augmentation, semi-supervised learning or self-supervised methods that we have seen before and that could help to improve the construction of models. However, integrating these methods into an AutoML tool is not simple since these techniques also have a large number of hyperparameters that need to be configured, which makes their adoption more difficult. This problem can be solved by using a different AutoML approach focused on automatically searching the best configuration from a set of given Machine Learning algorithms. This approach has been successfully implemented in tools like SMAC [80], Auto-WEKA [81], Auto-Sklearn [82] or AutoAlbument [83] for structured data but it has not been studied for unstructured data (like images or text). Due to these reasons, AutoML methods and tools are difficult to adopt in the life-science context, and it is necessary the development of new AutoML methods and tools that work with a limited amount of annotated images and resources.

Up to now, we have seen a lot of solutions that are focused on how to build and train a Deep Learning model; however, there is still a very important step remaining, that is how to use those trained models.

1.4 Challenge 3: Democratisation of models' usage

Currently, we can find a large number of trained models ready to be used[2]. However, the use of these models by life scientists without a Deep Learning background, or by users in general, is not straightforward. This is due to the fact that to use these models it is necessary to have some knowledge about programming and the framework where the model was trained. To avoid this situation, and bring Deep Learning models closer to the end users, the usual procedure is to embed these models into applications, allowing their use in a transparent way. Thus, we can find Deep Learning models in applications and devices that we use in a daily basis, such as smartphones, smartwatches or electronic devices in general [84]. These devices have space and resource restrictions, however traditional Deep Learning models are expensive in terms of computation, memory and power consumption, which make it difficult to integrate them into these devices [85]. Furthermore, in the biomedical field, the adoption of these models by life scientist is not straightforward since Deep Learning models should be integrated into bioimaging tools such as ImageJ [86], Icy [87] or CellProfiler [88]. Usually the connection between these tools and Deep Learning models is not easy due to the different characteristics of each application, or the language in which they were implemented. So, we can detect two major problems in the democratisation of Deep Learning models' usage: (1) the resource constraints of the electronic devices and (2) the difficulties of connecting these models with specific tools.

The first problem that we have mentioned is that Deep Learning models are not only data demanding, but also, computationally demanding during inference time (for instance, a VGG16 model takes 0.125 seconds to infer the result for a given image

[2] `https://huggingface.co/`

in a NVIDIA TX1 [89]). To solve this problem, and reduce the resources that are required when deploying and using a model in edge devices, a new kind of deep neural networks, called compact networks or hardware-aware networks, have been designed taking into account not only their accuracy, but also its computational complexity [90]. Initially, those networks were manually designed by pruning bigger networks [91], building networks based on operations that are cost-friendly [92], or applying a number of network compression techniques [93]. Those manual methods have been recently replaced by neural architecture search (NAS) techniques that automatically search for the most accurate and efficient architecture under memory and space constraints [90]. However, the performance in terms of accuracy of compact networks is usually lower than standard size networks, which can be a problem in fields where a high performance is necessary. Furthermore, most of these networks have been designed, trained and tested using large natural image datasets such as Imagenet [39], and it is worth studying whether they work well in life science tasks.

A different approach that tackle the problem of constrained resources is the use of techniques that reduce the size of standard-size networks while preserving their performance, such as pruning [91] or quantification [94]. Pruning is a technique that consists in eliminating connections and parameters of a network in order to reduce its size and maintain the precision of the network; whereas, model quantisation consists in converting the parameters of the network from floating point numbers to 4 or 8 bit integers, which greatly reduces the size of the network as well as increases its speed due to the fewer number of floating point operations that have to be performed.

Both compact deep networks, and pruning and quantisation methods have been successfully employed in the biomedical context for glaucoma detection [95], diabetic retinopathy diagnosis [96] or skin cancer classification [97]. Those models are usually trained by applying transfer learning and generally using manually designed compact networks such as ResNet-18 [95] or MobileNet [97]. However, automatically designed and quantized compact models are scarce in this context; probably, due to the fact that they are optimised for natural images from the ImageNet challenge; and, it is not clear whether these models can be properly transferred to biomedical images, or whether they obtain better results than manually designed compact networks. In addition, these networks are usually the ones that are largely forgotten when trying new techniques such as semi-supervised or self-supervised methods, since accuracy tends to prevail over efficiency. Something similar happens when we use AutoML tools to build Deep Learning models since these compact networks are not usually considered.

The second problem to be addressed for the democratisation of models' usage in life-science is their connection with bioimaging tools. Most bioimaging applications such as ImageJ [86], Icy [87] or ImagePy [98] have the common feature that they allow their users to make plugins to increase their functionality. There are several projects that connect a concrete bioimaging tool with a particular Deep Learning or Machine Learning library. The ImageJ-TensorFlow plugin [99] connects ImageJ with the Machine Learning framework TensorFlow [100]; similarly, CellProfiler [88] has been connected with both TensorFlow and Caffe [101]; another project that integrates Machine Learning techniques in ImageJ is Trainable Weka Segmentation [102]. Rapid Learning [103] is a similar project that integrates Machine Learning techniques into Icy. Also, KNIME [104], a data analytics platform, has included some Deep Learning

frameworks and models in its platform. These are some of the tools and plugins that connect biomedical tools with Deep Learning models. However, this approximation usually leads to the creation of ad-hoc plugins and not in a general approach that facilitates the connection with new models. Recently, a more general approach has emerged, called DeepImageJ [105], which allows the connection of Deep Learning models and tools for pixel and object classification, instance segmentation, denoising and virtual staining with ImageJ. However, this approach focuses exclusively on ImageJ; and it cannot be used in other bioimaging tools. Therefore, it is necessary the creation of a common procedure that allows the general connection of biomedical tools and Deep Learning models.

1.5 Objectives

Up to now, we have presented the context where we can frame our research: the democratisation of Deep Learning methods. Taking into account the challenges and problems posed in the previous sections, this work proposes a series of objectives that would help to reduce the amount of resources that are required to train Deep Learning models, and facilitate their construction and usage. Thus, the achievement of the following objectives will mark the lines of our work.

O1. Develop techniques that reduce the amount of annotated images and computational resources that are required to train and use Deep Learning models.

O2. Develop techniques and tools that allow users to automatically construct Deep Learning models without having expert knowledge.

O3. Develop techniques to reduce the size and computational resources required to apply Deep Learning techniques. In addition, these techniques have to allow the construction of new models capable of have a good performance on edge devices in various fields of life sciences.

O4. Implement libraries and tools that facilitate the connection of Deep Learning techniques with image analysis tools used in different fields. These libraries and tools must be simple and adapted to be used by life scientists without a Deep Learning background.

O5. Use the methods, libraries and tools developed in the previous objectives to tackle actual life science tasks to show the applicability and usefulness of our methods.

In the next chapter, we present how these objectives have been achieved, as well as the main results that we have obtained.

Chapter 2

Results and discussion

This chapter presents the results obtained in this work, and that have been collected in the different publications that make up this memoir. These results are the consequence of completing the objectives set at the end of the previous chapter. In Table 2.1, we can see a summary of our results together with their corresponding objective, and the publications wherein they have been presented. Each of these results and how they have been achieved are detailed below.

2.1 Objective O1: Reducing the amount of data resources

To achieve our first objective, which consists in reducing the amount of annotated images and computational resources that are required to train and use Deep Learning models, we have studied two different approaches that are: (1) developing techniques to increase the number of images available to train a model; and (2) studying techniques that allow the reduction of the number of annotated images that are needed to train a model.

2.1.1 Facilitating Data Augmentation

First, we have studied data augmentation techniques to increase the number of images available to train a given model [53]. There are numerous libraries that allow us to perform data augmentation in a simple way, such as Imgaug [112], Augmentor [113], Albumentations [114] and AutoAugment [115]. These libraries were initially developed by focusing on data augmentation for classification problems; however, in recent years they have increased their functionality allowing users to perform data augmentation for detection and segmentation task regardless of the underlying Deep Learning framework. By reviewing existing data augmentation tools, we found out that most of these tools were designed to perform data augmentation on classification problems for 2-dimensional images. In addition, each tool had different image augmentation techniques. Our work has been focused on allowing the use of data augmentation

N.	Results	Objective	Contributions
R1	Creation of a data augmentation library for object classification, localisation, detection, semantic segmentation and instance segmentation.	O.1	[106]
R2	Development of several semi-supervised methods to create Deep Learning models from a small number of images.	O.1	[107]
R3	Development of a Deep Learning library and an application that assists the user in the entire process of creating a Deep Learning model, from the annotation of images to the final creation of the model.	O.2	[107]
R4	Development of a library that facilitates the creation of networks adapted to edge-devices, using different techniques such as compact networks or quantification.	O.3	[108]
R5	Creation of a library that provides a common access point for classification models of several Deep Learning frameworks and several plugins that connect this API with different bioimaging tools.	O.4	[109]
R6	Resolution of different problems in the biomedical field using the techniques previously developed.	O.5	[110, 111]

Table 2.1: Results obtained during the completion of this work, together with the objectives reached with each result and the publication in which it was collected.

techniques on different problems. As a result, we have created a data augmentation library, called CLoDSA, that was the first image augmentation library for object classification, localisation, detection, semantic segmentation, and instance segmentation that worked not only with 2-dimensional images but also with multi-dimensional images. Currently, data augmentation tools have included more functionalities and most of them can be used for detection, semantic segmentation, and instance segmentation. However, CLoDSA is still the only one that allows users to work with videos as well as changing the annotation of an image. Furthermore, this library is independent of the format of the images to be augmented as well as the Deep Learning library that is going to be used to train the models.

CLoDSA augmentation procedure is flexible to adapt to different needs and it is based on six parameters: the dataset of images, the kind of problem, the input annotation mode, the output annotation mode, the generation mode, and the techniques to be applied. The dataset of images is given by the path where the images are located; and the kind of problem is either classification, localization, detection, semantic segmentation, instance segmentation, stack classification, stack detection, or stack segmentation (the former five can be applied to datasets of 2-dimensional images, and the latter 3 to datasets of multi-dimensional images). A summary of the different options for the input and output annotation mode can be seen in Table 2.2. The generation mode indicates how the augmentation techniques will be applied, linear or power. In the linear mode, given a dataset of n images, and a list of m augmentation techniques, each technique is applied to the n images producing $n \times m$ images. The power mode is a pipeline approach where augmentation techniques are chained together. In this approach, the images produced in one step of the pipeline are added to the dataset that will be fed in the next step of the pipeline producing a total of $(2^m - 1) \times n$ new images (where n is the size of the original dataset and m is the cardinal of the set of techniques of the pipeline). Finally, the last parameter is the set of augmentation techniques to apply. We can distinguish two types of techniques depending on whether they change the position of the object, position variant techniques, or not, position invariant techniques. The list

of techniques available in CLoDSA is given in Table 2.3, and a more detailed explanation of the techniques and the parameters to configure them is provided in the project webpage[1].

Data	Task	Input format	Output format
2D Images	Classification	A folder for each class of image	A folder for each class of image
			An HDF5 file
			A Keras generator
	Localization	Pascal VOC format	Pascal VOC format
			An HDF5 file
	Detection	Pascal VOC format	Pascal VOC format
		YOLO format	YOLO format
	Segmentation	A folder with masks and images	A folder with masks and images
			An HDF5 file
			A Keras generator
	Inst. Seg	COCO format	COCO format
		JSON format from ImageJ	JSON format from ImageJ
ND Images	Vid Class	A folder for each class of video	A folder for each class of video
	Vid Detect	Youtube BB format	Youtube BB format
	Stack seg	Tiff files with stack and mask	Tiff files with stack and mask

Table 2.2: List of supported annotations formats in CLoDSA.

Therefore, the first contribution of this work is an approach that allows anyone to automatically apply image augmentation techniques to several computer vision problems. Such a method was implemented in CLoDSA, the first library devoted to image augmentation for object classification, localization, detection and semantic segmentation.

2.1.2 Semi-supervised learning methods

With CLoDSA, we studied how to augment the size of a dataset of images for training Deep Learning models; in addition, we have also studied different techniques that allow us to reduce the number of annotated images that are needed to train a model. In particular, we have studied two different approaches, a classical approach using semi-supervised learning methods and transfer learning techniques, and a topological approach that combines semi-supervised learning with Topological Data Analysis.

Distillation approach

In the first approach, the techniques that we studied were semi-supervised learning methods, in particular, data and model distillation, and transfer learning. Data and model distillation are two forms of self-training [55], a special kind of semi-supervised

[1] `https://github.com/joheras/CLoDSA`

Position invariant techniques	Position variant techniques
Average blur	Crop
Bilateral blur	Elastic deformation
Brightness noising	Flip
Color noising	Rescale
Contrast noising	Rotation
Dropout	Skewing
Gamma correction	Translation
Gaussian blur	
Gaussian noise	
Hue jitter	
Median blur	
Normalization	
Random erasing	
Salt and pepper	
Saturation jitter	
Sharpen	
Value jitter	
Channel shift	
Lightning	
Change space color	

Table 2.3: List of augmentation techniques available in CLoDSA divided into two types of techniques depending on whether they change the position of the object, position variant techniques, or not, position invariant techniques.

learning. In the case of data distillation [116], given a model trained on manually labelled data, this technique applies such a model to multiple transformations of unlabelled data, ensembles the multiple predictions, and, finally, retrains the model on the union of manually labelled data and automatically labelled data. In the case of model distillation [117], several models are employed to obtain predictions of unlabelled data; subsequently, those predictions are ensembled, and used to train a new model. Both techniques can also be combined as shown in [116]. A simpler variant is plain distillation, which consists in using a single model trained on manually labelled data to obtain predictions of unlabelled data.

We also studied the application of these techniques into the biomedical field, whose images differ from the natural images generally used. Specifically, we carried out an exhaustive study of the behaviour of different models trained by combining a new semi-supervised method with transfer learning in 10 biomedical datasets with few annotated images, see Table 2.4. In particular, we developed an iterative process, see Figure 2.1. The process starts by training m models $\{M_1, \ldots, M_m\}$ using an annotated dataset (X, Y) where X is a set of images and Y the corresponding labels. We train each model by applying fine-tuning from the ImageNet challenge and following the two-stage procedure presented in [118] — in this way, we take advantage of the numerous

descriptors learned from the Imagenet dataset. In the first stage of the training process, we replace the last layers of the model (that is, the layers that give us the classification of the images), with new layers adapted to the number of classes of each particular dataset. Then, we train these new layers with the data of each particular dataset for two epochs. Since training only the last layers of a model may not be enough to obtain a good performance in the new dataset, it is necessary to conduct a second stage. In the second stage, we unfreeze the whole model and retrain all the layers of the model with the new data for several epochs – this parameter can be configured. When training all layers of the model, we should be careful with the learning rate used. The lower layers of the model have the most basic descriptors (colours, borders, shapes, ...), which are common for all images, and as we go up of our model, the descriptors become more specific to the used data. Thus, the idea is to modify the lowest descriptors minimally, and make a greater modification in the upper layers of the model. This is translated into using a low learning rate, lr, in the initial layers and a higher learning rate in the following layers. Then, to train our models, we use a learning rate slice $(\frac{lr}{100}, lr)$, that starts at a low learning rate $(\frac{lr}{100})$ for the lower layers and increases as we move through the layers until we reach lr in the higher layers. In addition, we look for this specific lr that improves the performance of our model; i.e., we select lr that decreases the loss to the minimum possible value using the approach presented in [119]. In particular, we select the learning rate with the lowest loss value, and, in case that this learning rate is too small ($lr < 1e^{-5}$), we change its value to $1e^{-3}$ – this parameter was experimentally fixed.

Dataset	Number of Images	Number of Classes	Description
Blindness [120]	3662	5	Diabetic retinopathy images
Chest X Ray [121]	2355	2	Chest X-Rays images
Fungi [122]	1204	4	Dye decolourisation of fungal strain
HAM 10000 [123]	10015	7	Dermatoscopic images of skin lesions
ISIC [124]	1500	7	Colour images of skin lesions
Kvasir [125]	8000	8	Gastrointestinal disease images
Open Sprayer [126]	6697	2	Dron pictures of broad leaved docks
Plants [127]	5500	12	Colour images of plants
Retinal OCT [121]	84484	4	Retinal OCT images
Tobacco [128]	3492	10	Document images

Table 2.4: Description of the biomedical datasets employed in our experiments.

Now, for each unlabelled image $\bar{x} \in \bar{X}$, where $\bar{X}$ is a dataset of unlabelled images, and using t image transformations $T = \{T_1, \ldots, T_t\}$, we generate $t + 1$ new images $T_0(\bar{x}), T_1(\bar{x}), \ldots, T_t(\bar{x})$, where T_0 is the identity. Subsequently, we apply each model M_i to each $T_j(\bar{x})$ and obtain as a result $(\bar{y}_{i,j}, \bar{p}_{i,j})$ where $\bar{y}_{i,j}$ is the class predicted by M_i for $T_j(\bar{x})$, and $\bar{p}_{i,j}$ is its associated confidence. After that, we ensemble the predictions $\{(\bar{y}_{i,j}, \bar{p}_{i,j})\}_{i \in [1,\ldots,m], j \in [0,\ldots,t]}$ by using the weighted majority voting scheme, where the weights are the confidence score of each prediction, and obtain $(\bar{y}, \bar{p})$. Finally, if $\bar{p}$ is over a fixed threshold, then we add $\{(\bar{x}, \bar{y})\}$ to (X, Y) and remove $\bar{x}$ from $\bar{X}$; and, the process is iterated. The process ends when there are not unlabelled images — this condition can be replaced by others, such as a maximum number of iterations,

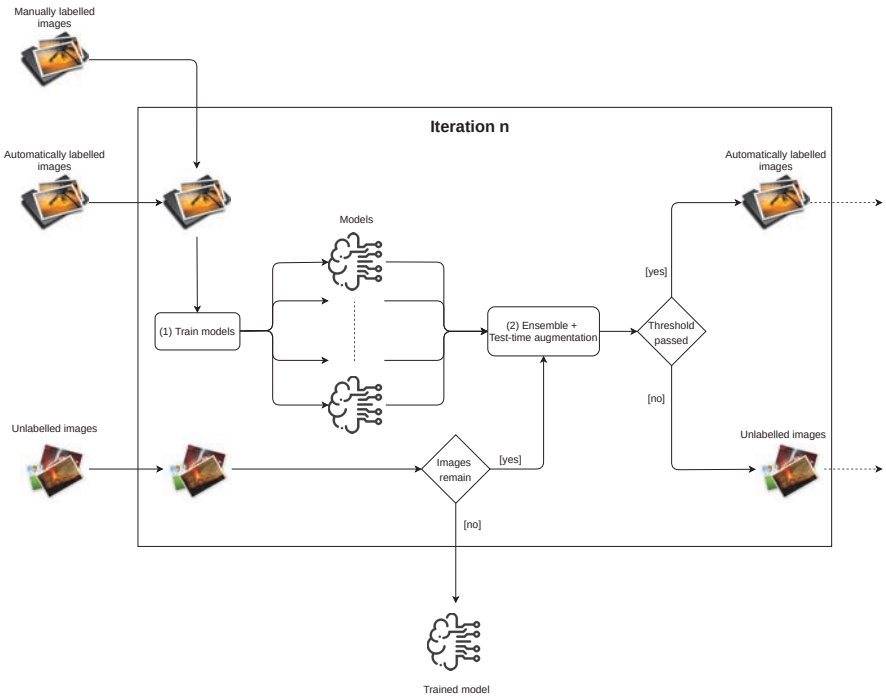


Figure 2.1: Workflow of an iteration of our semi-supervised learning method. **Inputs of the iteration**: set of manually labelled images, set of automatically labelled images and set of unlabelled images. **Outputs of the iteration**: set of automatically labelled images and set of unlabelled images. **Output of the process**: Trained model.

or a condition on model improvement. The result of this process is a model; in particular, the model with the best performance with respect to an independent labelled test set, in the last iteration. From this general process we defined 7 semi-supervised learning methods:

No Distillation (N.D.): This is the most basic case and occurs when there are not unlabelled images in the original dataset, that is, $\bar{X} = \emptyset$. Then, the process is reduced to train and select the best model model with respect to an independent test set.

Data Distillation without thresholding (D.D.): This case appears when, instead of having a set of base models, we only train one model ($M = \{M_1\}$), and we do not set a threshold; that is, the threshold value is 0. In this case, the process does not iterate, since all the unlabelled images are annotated in the first iteration.

Iterative Data Distillation (I.D.D.): As in the previous case, a single model is trained ($M = \{M_1\}$); but we establish a threshold to be passed, which leads us to have, prob-

ably, several iterations.

Model Distillation without thresholding (M.D.): This case starts by training a set of models; but, we do not perform test-time augmentation of the unlabelled images, that is, $(T = \{T_0\})$, and only ensemble the model predictions of the given image. In addition, the value of the threshold is set to 0; that is, we do not have an iterative process.

Iterative Model Distillation (I.M.D.): In this case, we also start by training a set of models, and as in the previous case, we do not perform test-time augmentation $(T = \{T_0\})$. The main difference with the previous case is that we establish a threshold, which leads us to have an iterative process.

Model + Data Distillation without thresholding (M.D.D): This case is very similar to the general workflow. We use a set of models and test-time augmentation to annotate the unlabelled images. The only difference is that the threshold is set to 0; and, therefore, we do not have an iterative process.

Iterative Model + Data Distillation (I.M.D.D): This is the general workflow explained previously.

For testing the aforementioned methods, we split our 10 datasets of the benchmark into two different sets: a training set with the 75% of images and a testing set with the 25% of the images. With this division, we performed an analysis of the seven processes explained before: No Distillation (N.D.), Data Distillation (D.D.), Iterative Data Distillation (I.D.D), Model Distillation (M.D.), Iterative Model Distillation (I.M.D.), Model + Data Distillation (M.D.D.) and Iterative Model + Data Distillation (I.M.D.D). For each process, we carried out three different experiments, starting from 25, 50 and 75 annotated images of the training set per class and considering the rest of the training images as unlabelled, we applied the seven processes; and, additionally, we applied the N.D. process to the whole dataset. Furthermore, for the processes that use test-time augmentation, we selected five augmentation techniques, namely, horizontal flip, vertical flip, horizontal and vertical flip, blurring, and gamma correction. In the model distillation processes, we used the architectures ResNet34, ResNet50, ResNet101 and DenseNet121. Finally, for the iterative processes, we established a threshold value of 0.8.

The result of these experiments can be seen in Table 2.5 and Figure 2.2, we can notice that the improvements regarding the ND method achieved by using our methods range from a 3% in the worst case, up to a 10% in the best case. We can also notice that the iterative versions of our processes produce better results than their non-iterative counterparts in all the cases but one. In addition, we can observe that as we increase the number of images initially annotated, the results considerably improve. Moreover, in some cases, it is possible to get results close to those obtained using the N.D. process applied to the whole dataset by using just a small part of the data.

	25 per class							50 per class							75 per class							Full
	N.D.	D.D.	I.D.D.	M.D.	I.M.D.	M.D.D.	I.M.D.D.	N.D.	D.D.	I.D.D.	M.D.	I.M.D.	M.D.D.	I.M.D.D.	N.D.	D.D.	I.D.D.	M.D.	I.M.D.	M.D.D.	I.M.D.D.	
Blindness	0.66	0.71	**0.73**	0.70	0.72	0.70	0.73	0.71	0.76	**0.78**	0.72	0.71	0.74	0.76	0.72	0.76	0.76	0.75	**0.76**	0.75	0.76	0.83
Chest X Ray	0.73	0.73	0.78	0.74	0.74	0.76	**0.83**	0.85	0.91	0.90	0.88	**0.92**	0.89	0.88	0.87	0.88	0.83	0.90	**0.91**	0.90	0.90	0.93
Fungi	0.74	0.69	0.73	0.74	0.74	0.74	**0.75**	0.80	0.82	0.79	0.82	0.83	**0.84**	0.83	0.90	0.87	0.89	0.91	0.91	0.91	**0.92**	0.96
HAM10000	0.55	0.61	**0.65**	0.60	0.63	0.63	0.63	0.63	0.67	**0.72**	0.64	0.66	0.65	0.66	0.64	0.69	**0.74**	0.69	0.72	0.69	0.74	0.88
ISIC	0.74	0.77	0.78	0.78	**0.83**	0.80	0.83	0.81	0.81	0.84	0.82	0.83	0.82	**0.85**	0.84	0.83	0.85	0.85	**0.87**	0.84	0.87	0.87
Kvasir	0.79	0.85	0.88	0.88	0.89	0.88	**0.89**	0.84	0.86	0.88	0.88	**0.90**	0.88	0.90	0.87	0.90	0.91	0.90	0.91	0.90	**0.91**	0.93
Open Sprayer	0.84	0.90	0.91	0.83	0.84	0.85	**0.93**	0.87	0.86	0.90	0.90	0.91	0.91	**0.92**	0.90	0.92	0.94	0.94	0.94	0.94	**0.94**	0.97
Plants	0.83	0.86	0.89	0.88	**0.91**	0.88	0.91	0.89	0.91	**0.93**	0.93	0.93	0.92	0.93	0.91	0.92	0.93	0.94	**0.95**	0.93	0.95	0.96
Retinal OCT	0.90	0.90	0.86	0.93	**0.96**	0.93	0.94	0.93	0.95	0.95	0.96	**0.98**	0.97	0.97	0.94	0.97	0.93	0.98	**0.99**	0.98	0.98	0.99
Tobacco	0.66	0.69	0.70	0.74	**0.76**	0.72	0.74	0.72	0.75	0.77	0.77	**0.80**	0.79	0.76	0.78	0.81	**0.84**	0.81	0.81	0.81	0.79	0.86
Mean	0.74	0.77	0.79	0.78	0.80	0.79	0.82	0.81	0.83	0.85	0.83	0.85	0.84	0.85	0.84	0.86	0.86	0.87	0.88	0.87	0.88	0.92
S.D.	0.10	0.10	0.09	0.10	0.10	0.10	0.10	0.09	0.09	0.08	0.10	0.10	0.10	0.10	0.10	0.08	0.07	0.09	0.09	0.9	0.08	0.05
z (Wilcoxon)		-1.91	-2.40	-2.49	-2.53	-2.67	-2.81		-2.57	-2.71	-2.82	-2.67	-2.84	-2.83		-1.96	-1.75	-2.84	-2.84	-2.69	-2.84	
p		0.056	0.016	0.013	0.011	0.008	0.006		0.010	0.007	0.005	0.008	0.004	0.005		0.051	0.080	0.004	0.004	0.007	0.004	
r		-0.43	-0.54	-0.56	-0.57	-0.60	-0.63		-0.57	-0.61	-0.63	-0.60	-0.64	-0.63		-0.44	-0.39	-0.64	-0.64	-0.60	-0.64	

Table 2.5: Comparison of the performance of the seven different processes (N.D.: No distillation, D.D.: Data distillation, I.D.D.: Iterative Data distillation, M.D.: Model distillation, I.M.D.: Iterative Model distillation, M.D.D.: Model + Data distillation, I.M.D.D.: Iterative Model + Data distillation) in 10 datasets with 25, 50 and 75 annotated images per class. The "Full" column indicates the accuracy of the N.D. method applied to the whole dataset. The best results are highlighted in bold face. Wilcoxon signed-rank tests for comparing the results obtained by the N.D process with respect to the other six processes in each block of 25, 50 and 75 annotated images per class are also included.

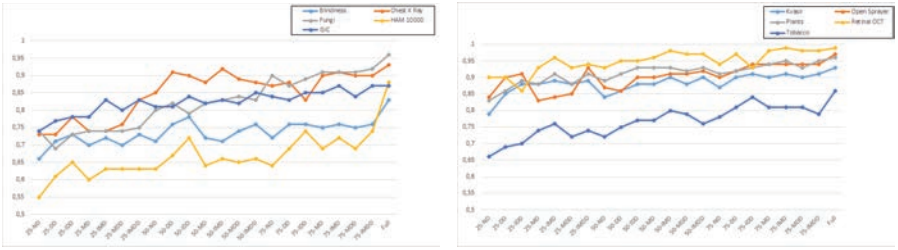


Figure 2.2: Comparison of the performance of the seven different processes (N.D.: No distillation, D.D.: Data distillation, I.D.D.: Iterative Data distillation, M.D.: Model distillation, I.M.D.: Iterative Model distillation, M.D.D.: Model + Data distillation, I.M.D.D.: Iterative Model + Data distillation) in 10 datasets with 25, 50 and 75 annotated images per class.

Hence, the second contribution of this work is a semi-supervised learning method that allows the use of deep learning techniques to solve an image classification problem with few resources. In particular, this method, that combines transfer learning and semi-supervised learning, allows users to train deep models with small, and partially annotated datasets of images. Furthermore, our semi-supervised learning method improves the accuracy of models up to a 10% when working with partially annotated datasets.

Topological approach

In our second approach to reduce the amount of annotated data that is necessary to train Deep Learning models, we have started to explore techniques from Topological Data Analysis (TDA) [129], a field that extracts topological and geometrical information from data, to define different semi-supervised learning methods to solve binary classification problems. In particular, we created several semi-supervised learning meth-

ods following two different topological approaches. We assume some familiarity with notions employed in TDA such as Vietoris-Rips filtration (we denote by V_X to the Vietoris-Rips filtration associated with a set X), persistence diagrams (we denote by $P(F)$ to the persistence diagram associated with a filtration F), and the bottleneck and Wasserstein distances (denoted by d_B and d_W respectively). For a detailed introduction to these topics see [129].

An important concept in TDA is the Manifold Hypothesis [130], that considers that high dimensional data tends to lie in low dimensional manifolds, and that inspired our definition of a semi-supervised learning method for binary classification tasks. Intuitively, our method is based on the idea that given two sets of data points A and B, we can define two manifolds associated with each set, $\mathcal{M}_A$ and $\mathcal{M}_B$ respectively. Now, given an unlabelled data point x that belongs to either A or B; if x belongs to A, analogously for B, then the manifold associated with $A \cup \{x\}$ and $\mathcal{M}_A$ will be more similar than the manifold associated with $B \cup \{x\}$ and $\mathcal{M}_B$. Then, we start with a set X_1 of points from class 1, a set X_2 of points from class 2, and a set X of unlabelled points. The objective of our algorithms is to annotate the elements of X by using topological properties of X_1 and X_2. In particular, our semi-supervised learning algorithm takes as input the sets X_1 and X_2, a point $x \in X$, a threshold value t, and a flag that indicates whether the bottleneck or the Wasserstein distance should be used, we denote the chosen distance as d. The output produced by our algorithm is whether the point x belongs to X_1, X_2 or none of them. In order to decide the output of the algorithm, our hypothesis is that if a point belongs to X_1, analogously for X_2, the topological variation that X_1 will suffer when adding the point will be minimal; whereas if the point does not belong to X_1, the variation will be greater. In particular, we proceed as follows:

1. Construct the Vietoris-Rips filtrations V_{X_1}, V_{X_2}, $V_{X_1 \cup \{x\}}$ and $V_{X_2 \cup \{x\}}$;
2. Construct the persistence diagrams $P(V_{X_1})$, $P(V_{X_2})$, $P(V_{X_1 \cup \{x\}})$ and $P(V_{X_2 \cup \{x\}})$;
3. Compute the distances $d(P(V_{X_1}), P(V_{X_1 \cup \{x\}}))$ and $d(P(V_{X_2}), P(V_{X_2 \cup \{x\}}))$, from now on d_1 and d_2 respectively;
4. If both d_1 and d_2 are greater than the threshold t, return none; otherwise, return the set associated with the minimum of the distances d_1 and d_2.

The above algorithm is diagrammatically described in Figure 2.3, and it is applied for all the points of the set of unlabelled points X. Note that if we use a threshold value of 0, the algorithm will annotate all the points of X; however, this might introduce some noise.

In the second approach, we looked at the connectivity of the data. In particular, we focused on the minimum radius that the Vietoris-Rips complex associated with a set has to take to be connected. As in the previous case, we start with a set X_1 of points from class 1, a set X_2 of points from class 2, and a set X of unlabelled points. Our semi-supervised learning algorithm takes as input the sets X_1 and X_2, a point $x \in X$. The output produced by our algorithm is whether the point x belongs to X_1, X_2 or

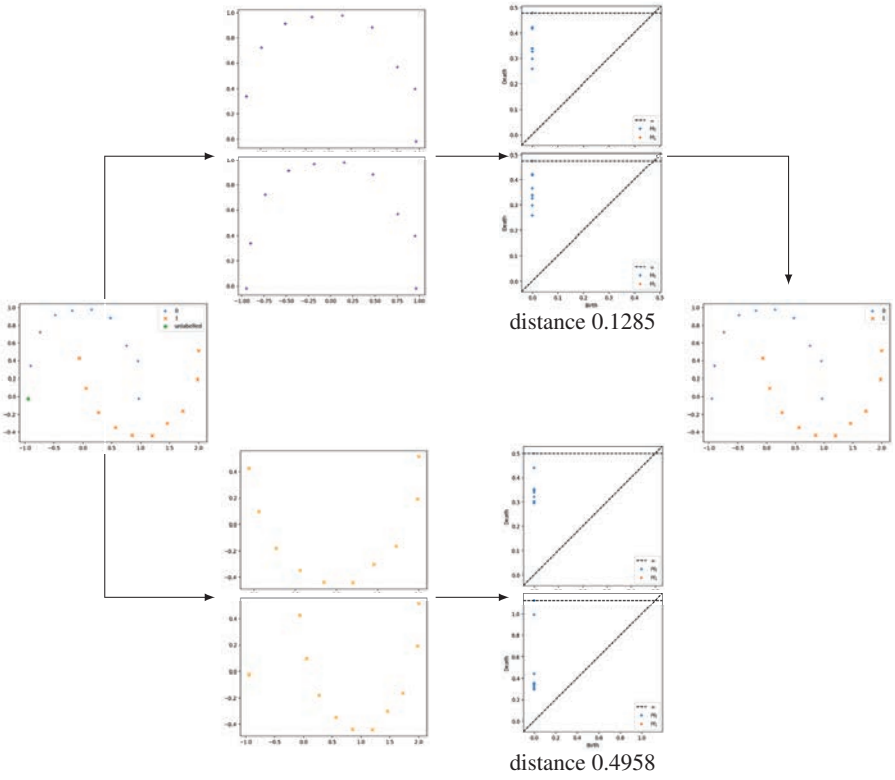


Figure 2.3: Example of the application of our homological semi-supervised method using the bottleneck distance, and using 0.6 as threshold value.

none of them. In order to decide the output of the algorithm, our hypothesis is that if a point belongs to X_1, analogously for X_2, the minimum connectivity radius of the associated Vietoris-Rips complex does not change considerably; on the contrary, if the point does not belong to the set X_1, analogously for X_2, the radius will increase. In particular, we proceed as follows:

1. Construct the Vietoris-Rips complex V_{X_1}, V_{X_2}, $V_{X_1 \cup \{x\}}$ and $V_{X_2 \cup \{x\}}$;

2. Compute the minimum connectivity radius $r(V_{X_1})$, $r(V_{X_2})$, $r(V_{X_1 \cup \{x\}})$ and $r(V_{X_2 \cup \{x\}})$, from now on r_1, r_2, r_1' and r_2' respectively;

3. Compute the radius variation $|r_1 - r_1'|$ and $|r_2 - r_2'|$ from now on d_1 and d_2 respectively;

4. If both d_1 and d_2 are zero, return none; otherwise, return the set associated with the minimum of the differences d_1 and d_2.

The above algorithm is diagrammatically described in Figure 2.4, and it is applied for all the points of the set of unlabelled points X.

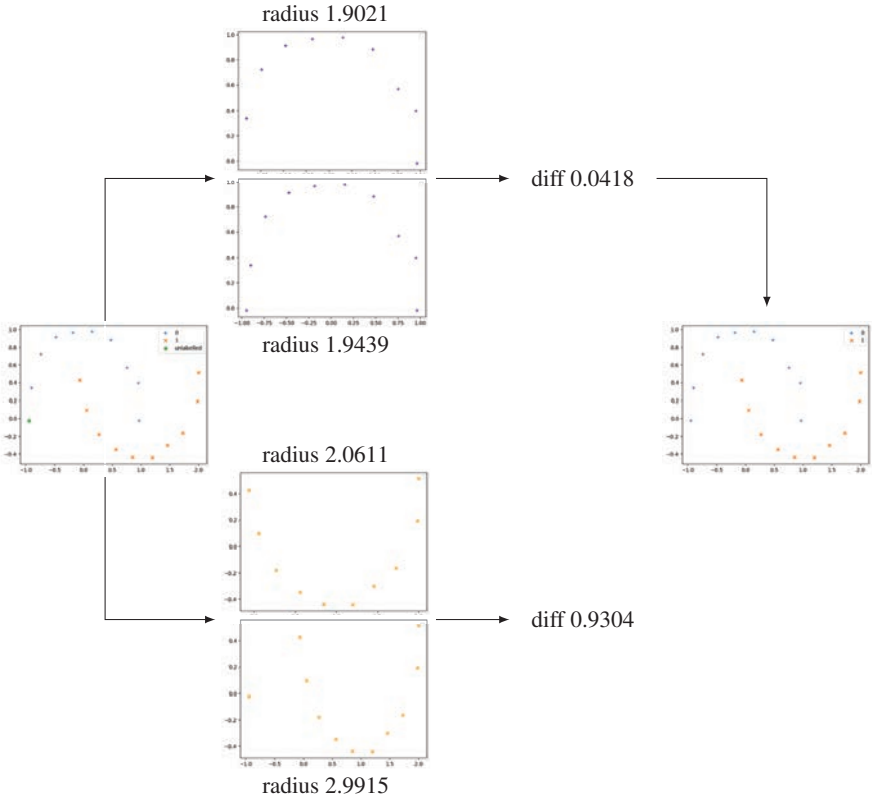


Figure 2.4: Example of the application of our connectivity semi-supervised method.

We carried out a thorough analysis of the developed methods by using 3 synthetic datasets, 5 structured datasets, and 2 datasets of images. The results show that the semi-supervised methods developed using a homological annotation method improve both the base results and those obtained with the classical semi-supervised learning methods like pseudo-labelling [58], reaching improvements of up to a 16%. Nevertheless, this work has only been tested on small datasets, so it is necessary to see how it behaves in real problems with larger datasets, a task that remains as further work.

Therefore, the third contribution of this work is twofold: a semi-supervised learning method using a homological annotation method that allows the use of Deep Learning and Machine Learning techniques to solve a binary classification problem with a small amount of data, and a connectivity method wit the same aim. In particular, these methods, that use TDA techniques and are based in the homological properties of the data reaches improvements of up to a 16% in structured datasets.

2.2 Objective O2: Democratising Deep Learning models' construction

For our second objective, we aimed to democratise the construction of Deep Learning models. Specifically, we developed a set of techniques and tools that allow the automatic and simple construction of Deep Learning models without having expert knowledge. The solutions that exist to deal with this problem are AutoML tools, such as AutoKeras, Google AutoML or AutoGluon among others. However, most of these tools are based on Network Automatic Search (NAS) methods, which require a lot of computational resources, and also expert knowledge to be able to obtain good results, see Table 2.6. Our idea was to use the combination of transfer learning and semi-supervised learning methods developed to tackle Objective O1, to create accurate models in a simple way. In particular, we have created an AutoML tool based on semi-supervised methods that assists the user throughout the process of creating a model.

Our AutoML tool, called ATLASS, allows the user to annotate the images in a simple way using a graphical user interface (GUI), implemented in Java, that provides all the necessary features to annotate a dataset of images. This includes the functionality to visualise and organise the images by categories. In addition, the GUI includes a wizard that allows the users to configure the training process, choose the model to be trained, and also the desired training method. In addition, to help non-expert users, by-default options have been set to allow them to create accurate models automatically. With all this information, the tool creates a Jupyter notebook that can be executed locally or in the cloud (for instance, using Google Colab [2]) to obtain the trained model. This tool has been tested on both small datasets and partially annotated datasets, obtaining better results than the existing AutoML tools in terms of accuracy and training times. In particular, in Table 2.7, we can see a comparison of 5 different AutoML tools (AutoKeras [74], Devol [131], Ludwig [132], WndCharm [133] and AutoGluon [75]) with our methods, that are the semi-supervised learning methods presented in Section 2.1.2, using 11 small annotated biomedical datasets presented in [134]. From these results, we can conclude that our AutoML tool greatly outperforms other tools when working with small datasets in both, accuracy and speed. This is mainly due to the fact that AutoKeras, Devol, Ludwig, and AutoGluon employ Neural Architecture Search algorithms, a family of techniques that require large corpora of data to be trained; and WndCharm employs manually engineered features, that are fixed for all the dataset independently of the concrete problem. In contrast, our method is adapted to each particular problem, and applies transfer learning to reuse the knowledge learned from big datasets.

In summary, the fourth contribution of this work is a general AutoML method that combines transfer and semi-supervised learning techniques. In addition, we have proven that our AutoML method outperforms other AutoML tools both in terms of accuracy and speed when working with small datasets. Finally, we have developed an open-source tool, that allows users to annotate a dataset, and use it for training a model with our method in an easy way. Altogether, our approach simplifies the construction of fairly good image classification models in the biomedical context when working

[2] `https://colab.research.google.com`

	Open-source	Cloud-based	Framework	Language	Supports	Techniques
AutoKeras	Yes	No	Keras	Python	CNN, RNN and LSTM	NAS
AutoGluon	Yes	No	MxNet	Python	CNN, RNN	NAS
Google AutoML	No	Yes	TensorFlow	Python	CNN, RNN and LSTM	NAS and Reinforcement Learning
Devol	Yes	No	-	Python	CNN, RNN	NAS and Hyperparameter optimization
Ludwig	Yes	Yes	PyTorch	Python	CNN, RNN	NAS
WndCharm	Yes	No	-	Python and C++	CNN, RNN	Manually engineered features
ATLASS	Yes	No	Fastai	Python	CNN, RNN and compact networks	Semi-supervised learning and transfer learning

Table 2.6: Comparison of the characteristics of the main AutoML tools for Deep Learning.

	Binu.	CEle.	Cho	Hela	Liver aging	Liver (AL)	Liver (CR)	Lymp.	Pollen	RNAI	Term.	Mean (S.D.)	Mean Time (min)
AutoKeras	0.63	0.66	0.91	0.78	0.91	0.69	**1.00**	0.73	0.94	0.52	0.49	0.75 (0.17)	30
Devol	0.73	0.42	0.61	0.38	0.33	0.81	0.71	0.56	0.56	0.20	0.33	0.51 (0.19)	16
Ludwig	0.54	0.48	0.64	0.51	0.33	0.65	0.90	0.57	0.58	0	0.50	0.52 (0.22)	31
WndCharm	**1.00**	0.60	0.95	0.85	0.92	**1.00**	0.97	0.79	0.96	0.68	0.50	0.84 (0.17)	53
AutoGluon	0.46	0.80	**0.97**	0.88	0.71	0.94	0.71	0.57	0.84	0.36	0.55	0.71 (0.20)	10
Ours	**1.00**	**1.00**	**0.97**	**0.99**	**0.98**	**1.00**	**1.00**	**0.95**	**0.97**	**0.77**	**0.73**	0.94 (0.10)	16

Table 2.7: Comparison of the performance of AutoKeras, Devol, Ludwig, WndCharm, AutoGluon, and our AutoML method for automatically constructing models for 11 image classification problems using the accuracy metric. The last column provides the mean time required for training the models with each tool. The best results are highlighted in bold face.

with small, or partially annotated datasets.

2.3 Objective O3: Reducing the amount of computational resources

The next step in our work was to reduce the computational resources required to train and use Deep Learning models. Specifically, for Objective O3, we analysed techniques to reduce the size and computational resources required to apply Deep Learning techniques, allowing the construction of new models capable of having a good performance on edge devices in various fields of life sciences. Our solution to this problem was the study of compact networks, and methods like pruning and quantification. However, it is known that when using these networks and techniques, we might lose some accuracy with respect to standard-size networks [90, 92]. Our idea to solve this problem was to combine compact networks, both manually and automatically designed, with semi-supervised learning and transfer learning methods. This approach allowed us to obtain compact networks with a similar performance to standard size networks in biomedical tasks with a small amount of annotated images. In particular, we used the semi-supervised methods developed for the previous objectives, see Section 2.1.2, to train these compact networks. To do this, we carried out an exhaustive study of this combination of techniques using 10 partially annotated biomedical datasets, described in Table 2.4, and evaluate the performance of deep learning models and semi-supervised methods using such a benchmark.

For our study, we have split each of the datasets of the benchmark into two different sets: a training set with the 75% of images and a testing set with the 25% of the images. In addition, for each dataset we have selected 75 images per class using them

as labelled images and leaving the rest of the training images as unlabelled images to apply the semi-supervised learning methods. The splits used in our experiments and more information about datasets are available in the project webpage [3]. With this benchmark we studied the performance of 3 standard-size networks, 3 automatically designed compact networks, 4 manually designed networks, and 2 quantified networks presented in Table 2.8.

Name	Params (M)	FLOPs (M)	Top-1 acc (%)	Top-5 acc (%)	Design
ResNet-50	26	4100	76.0	93.0	Manual
ResNet-101	44	8540	80.9	95.6	Manual
EfficientNet-B3	12	1800	81.6	95.7	Auto
FBNet	9.4	753	78.9	94.3	Auto
MixNet	5	360	78.9	94.2	Auto
MNasnet	5.2	403	75.6	92.7	Auto
MobileNet v2	3.4	300	74.7	92.5	Manual
ResNet-18	11	1300	69.6	89.2	Manual
SqueezeNet	1.3	833	57.5	80.3	Manual
ShuffleNet v2	5.3	524	69.4	88.3	Manual
ResNet-18 quantized	11	-	69.5	88.9	Quantized
ResNet-50 quantized	26	-	75.9	92.8	Quantized

Table 2.8: Features of the architectures employed in this work. We measure the number of parameters (in millions), the FLOPs (in millions), and the Top-1 and Top-5 accuracy for the ImageNet challenge. In addition, we include how these architectures were designed. Quantized networks change the floating point parameters of the standard version for integer parameters, therefore they have the same number of parameters but do not perform floating point operations.

The results presented in Table 2.9 show that, with the combination of compact networks and semi-supervised methods, we can create compact models that are not only as accurate as bigger models, but also faster and lighter. In particular, we noticed that when training a model without using semi-supervised methods there are compact networks such as FBNet [90], MixNet [135], MNasNet [136] and ResNet-18 [137] that obtain a similar performance to standard size networks. Also, when we applied semi-supervised learning methods, specifically, the Plain Distillation method and the Data Distillation method introduced in Section 2.1, those models outperform the standard size models. In particular, the best results were obtained when we applied Data Distillation to MixNet and Plain Distillation to ResNet-18. Another conclusion that we drew from our study was that, in general, automatically designed networks obtain better results than manually designed networks and quantized networks, with the exception of ResNet-18 that obtained similar results than automatically designed networks. Regarding the question of which semi-supervised method produced the best results for each kind of architecture, see Table 2.10, we concluded that the Data Model Distillation method is the best option for standard size networks; that Data Distillation is the best for automatically designed networks; and, in the rest of cases, there is no a general rule, although the Data Distillation approach generally obtains good results.

[3] https://github.com/adines/SemiCompact

	Blindness	Chest X Ray	Fungi	HAM 10000	ISIC	Kvasir	Open Sprayer	Plants	Retinal OCT	Tobacco	Mean(std)
ResNet-50	59.3	89.9	**91.0**	**54.3**	87.6	**89.0**	91.3	84.3	97.4	**81.8**	**82.5(13.5)**
ResNet-101	58.2	**90.7**	86.9	52.0	84.0	83.8	95.8	84.3	96.4	80.1	81.2(14.1)
EfficientNet	53.6	84.1	84.7	52.8	85.0	85.4	**96.8**	84.0	98.1	72.9	79.7(14.8)
FBNet	57.5	87.4	89.0	47.2	85.2	88.9	95.4	81.8	94.9	73.3	80.1(15.3)
MixNet	**61.8**	89.5	89.7	46.9	**89.9**	86.8	95.5	**86.2**	**98.9**	76.7	82.2(15.3)
MNasNet	56.2	89.2	90.3	55.8	81.9	84.6	95.7	82.5	97.4	75.3	80.9(13.9)
MobileNet	52	86.9	89.0	46.7	84.1	82.1	89.1	82.9	91.0	69.4	77.3(15.1)
ResNet-18	56.3	90.3	94.2	53.7	86.8	84.1	91.6	80.0	97.7	77.5	81.2(14.4)
SqueezeNet	50.3	88.3	79.3	43.6	76.8	80.1	90.9	78.9	93.2	75.5	75.7(15.5)
ShuffleNet	39.5	85.7	69.9	37.6	78.9	67.0	89.6	51.9	33.9	40.7	59.5(20.2)
ResNet-18 quantized	45.1	77.8	88.1	47.0	86.5	84.2	91.3	75.1	91.6	55.8	74.3(17.2)
ResNet-50 quantized	48.6	77.2	83.2	42.9	78.6	81.1	85.4	77.7	91.6	69.7	73.6(15.0)

Table 2.9: Mean (and standard deviation) F1-score for the standard size models (ResNet-50, ResNet-101 and EfficientNet), compact models (FBNet, MixNet, MNasNet, MobileNet, ResNet-18, SqueezeNet, and ShuffleNet) and quantized models (ResNet-18 quantized and ResNet-50 quantized) for the base training method on the 10 biomedical datasets. The best result is highlighted in bold face.

	ResNet-50	ResNet-101	EfficientNet	FBNet	MixNet	MNasNet	MobileNet	ResNet-18	SqueezeNet	ShuffleNet	ResNet-18 quantized	ResNet-50 quantized
Base	82.6(13.5)	81.2(14.1)	79.7(14.8)	80.1 (15.3)	82.2(15.3)	80.9(13.9)	77.3(15.1)	81.2(14.4)	75.7(15.5)	59.5(20.2)	74.3(17.2)	73.6(15.0)
Plain	83.6(13.4)	82.9(13.3)	82.8(13.3)	82.6(12.8)	83.8(13.3)	82.6(14.2)	79.1(17.6)	84.2(13.8)	81.4(14.9)	57.8(22.7)	77.0(15.8)	70.7(21.2)
Data	83.1(13.7)	83.0(13.2)	83.1(14.5)	83.2(12.2)	**84.7(12.6)**	83.5(12.7)	81.4(15.4)	82.5(15.5)	81.7(13.7)	58.7(21.9)	77.5(13.5)	74.4(17.7)
Model	83.3(14.3)	83.0(14.2)	81.0(15.2)	80.5(15.0)	80.8(12.7)	77.9(14.8)	81.2(14.9)	82.9(14.4)	79.0(16.9)	56.2(23.9)	56.2(33.5)	62.7(26.7)
DataModel	83.7(14.0)	82.7(15.2)	82.0(15.7)	80.7(14.6)	80.8(14.2)	80.0(13.9)	80.5(15.9)	82.8(14.8)	79.2(16.6)	57.5(23.1)	53.1(32.1)	61.8(26.5)
FixMatch	64.2(20.0)	40.1(25.2)	55.4(26.4)	60.7(22.0)	74.4(25.1)	76.2(22.3)	74.8(22.9)	81.2(15.0)	52.0(18.3)	53.3(21.2)	74.4(22.9)	78.4(18.0)
MixMatch	47.1(28.8)	51.0(23.2)	64.3(22.6)	69.3(17.5)	49.1(28.9)	79.4(12.9)	68.1(25.6)	51.5(36.8)	57.7(19.4)	56.4(20.4)	61.7(25.9)	64.5(17.5)

Table 2.10: Mean (and standard deviation) F1-score for the different studied models and applying several semi-supervised methods. Methods: Base training (Base), Plain Distillation (Plain), Data Distillation (Data), Data Model Distillation (DataModel), FixMatch procedure (FixMatch), and MixMatch procedure (MixMatch). The best result is highlighted in bold face.

Finally, we studied the efficiency of the different networks, see Table 2.11. In particular, we analysed the model size, the time that takes each model to complete a training epoch, and the time that takes each model when applied for inference with a given image. From the results presented in Table 2.11, we can see that there is a great difference in size between the standard size and compact models. In particular, the difference in size between compact and standard size networks ranges from 30% to 97%, standing out the ShuffleNet network, however the accuracy of this model is usually lower than the rest. We can also notice that the quantized models reduce the size of original models by almost a 90%. Another important point when we test the efficiency of a model is the training time per epoch. In our experiments, using a Nvidia RTX 2080 Ti GPU with 11 GB RAM, standard size networks took approximately between 3 and 4 minutes per epoch. On the contrary, compact networks took less than 100 seconds per epoch (being MixNet the only exception). Finally, we studied the inference time of each model. We calculated the time that takes each model to infer the class of an image in a Intel(R) Xeon(R) W-2145 CPU with 3.70GHz, 16 CPUs cores and 32 GB. While standard size models took between 200 and 300 ms to classify an image, the compact networks take, at maximum, half the time. This is even more prominent in the case of ResNet-18, SqueezeNet, ShuffleNet, ResNet-18 quantized, and ResNet-50 quantized, that take, at least 4 less time than standard size networks. In particular, quantized versions of the models tool between 60% and 80% less time than their standard size versions.

Network	Size (MB)	Train Time (s)	Inference Time (ms)
ResNet-50	294	151	231
ResNet-101	512	263	238
EfficientNet	124	210	301
FBNet	42	78	121
MixNet	67	174	222
MNasNet	36	64	84
MobileNet	41	70	109
ResNet-18	135	51	63
SqueezeNet	15	52	57
ShuffleNet	0.36	15	14
ResNet-18 quantized	11	40	25
ResNet-50 quantized	23	100	44

Table 2.11: Efficiency of standard size models (ResNet-50, ResNet-101 and EfficientNet), compact models (FBNet, MixNet, MNasNet, MobileNet, ResNet-18, SqueezeNet, and ShuffleNet) and quantized models (ResNet-18 quantized and ResNet-50 quantized). We measure size (in MB), the training time per epoch (in seconds) and the inference time per image (in milliseconds).

In summary, the fifth contribution of this work is an exhaustive study that allow us to conclude that by applying the Data Distillation method to MixNet, or Plain Distillation to ResNet-18 we obtained models as accurate as standard size models but, also, faster and lighter. In addition, to facilitate the application of the methods studied in this work, we developed a library that simplifies the construction of compact models using semi-supervised learning methods.

2.4 Objective O4: Democratising Deep Learning models' usage

The next objective set for this work was related to the democratisation of the models' usage, specifically, for Objective O4, we aimed to facilitate the connection between Deep Learning models and bioimaging tools. We studied the different solutions given in the literature, and we saw that they consisted in ad-hoc plugins that only solve specific problems. Our idea was to create a general solution that allows the use of different Deep Learning models in different bioimaging tools. As a result, we created DeepClas4Bio, a project that aimed to facilitate the interoperability of bioimaging tools with Deep Learning frameworks. In particular, we developed an extensible API that provides a common access point for classification models of several Deep Learning frameworks, see Figure 2.5. This project groups the main Deep Learning frameworks, namely, Keras [138], Caffe [139], DeepLearning4J [140], MxNet [141] and PyTorch [142]. The users can use the pretrained models included in the API or they can train their own models and load them in the API to use them in a simple way. In addition, we created plugins for each of the most important biomedical tools such as

ImageJ, Icy and ImagePy, which allow users to easily use this API.

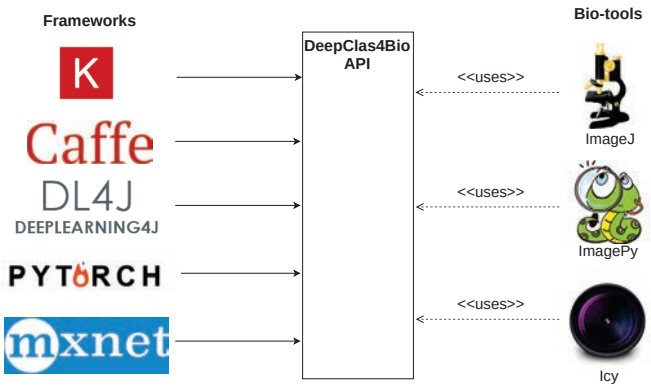


Figure 2.5: DeepClas4Bio is an API that sits in between Deep Learning frameworks and pre-trained models, and common bioimaging tools. It enables users to employ those models to make predictions on new images with minimal code from model developers and no code from the end user.

The main difference between DeepClas4Bio and other libraries, see Table 2.12, is that DeepClas4Bio is not focused on a specific task, framework or tool. As we can see in Table 2.12, RapidLearning [103] was developed in the RapidMiner [143] framework and it is only possible to use it with the Icy tool, while KNIME [104] is available for the Keras and TensorFlow frameworks, but its functionalities are only accessible through ImageJ. Instead, DeepClas4Bio is created as an extensible API that allows the connection between different Deep Learning frameworks and bioimaging tools through the creation of plugins. Our API is independent of the framework and the tool to use. DeepImageJ is a recent solution, created in 2021, that, as DeepClas4Bio, is independent of the framework to be used, however its use is restricted to ImageJ.

	RapidLearning	KNIME	DeepImageJ	DeepClas4Bio
Open-source	Yes	Yes	Yes	Yes
Year	2013	2006	2021	2019
Framework	RapidMiner (ML)	Keras and TensorFlow	Independent	Independet
Language	Java	Java	Python	Python
Bioimage tools	Icy	ImageJ	ImageJ	Independent

Table 2.12: Comparison of the characteristics of the main tools that try to connect Deep Learning techniques with bioimaging tools.

Hence, the sixth contribution of this work is a framework called DeepClas4Bio. It is a free and open-source project that allows the collaboration of bioimaging tools with image classification models developed in deep learning frameworks. This project has been successfully employed to construct several plugins in different bioimaging tools. Thanks to the DeepClas4Bio API, and the methodology developed to integrate it in the

workflow of life scientists' research, the development efforts can be greatly reduced when creating new tools for bioimaging that use deep learning classification models.

2.5 Objective O5: Applying developed methods

Finally, in order to verify that the methods developed during this work contribute to the life-science community, and, specifically, to the biomedical field, we have applied the developed methods to two actual biomedical problems proposed by life-scientists. In particular, we have applied the developed techniques to: (1) measure bacteria spread on motility images, and (2) predict epiretinal membrane (ERM) from retinal fundus images.

2.5.1 Measuring Bacteria Spread

Infectious diseases produced by antimicrobial resistant microorganisms are a major threat to human and animal health worldwide [144]. This problem is increased by the virulence and spread of these bacteria. Surface motility has been regarded as a pathogenicity element because it is essential for many biological functions, but also for disease spreading; hence, investigations on the motility behaviour of bacteria are crucial to understand chemotaxis, biofilm formation and virulence in general. To identify a motile strain in the laboratory, the bacterial spread area is observed on media solidified with agar. Up to now, the task of measuring bacteria spread was a manual, and, therefore, tedious and time-consuming task. In collaboration with the Centre of Biomedical Research of La Rioja, we addressed the problem of measuring bacteria spread on motility images by creating an automatic pipeline based on Deep Learning models. Such a pipeline consists of a classification model, with a F1-score of 99.85%, to determine whether the bacteria has spread to cover completely the Petri dish and a segmentation model, with a Dice coefficient of 95.66%, to determine the spread of those bacteria that do not fully cover the Petri dishes, see Figure 2.6.

To create the classification model, we fine-tuned several convolutional neural networks pretrained on the ImageNet dataset [39]; namely, the last layer of the convolutional networks was replaced with a sequence of linear layers where batch normalisation, dropout and a ReLU activation function were applied. In our experiments, we trained two ResNet architectures (Resnet 50 and 101), an EfficientNet architecture (EfficientNet B3), and a FBNet architecture. For each architecture, we constructed three models by using two different input image sizes (224×224 and 512×512), and using the progressive resizing approach – in this approach, we first trained the model using as input images of size 224×224, and, then, used the resulting model as a basis to train another model using images of size 512 × 512. We can notice from the results presented in Table 2.13 the benefits of using resizing since all the models were improved by using this approach.

To create the segmentation model, we fine-tuned several deep-learning segmentation algorithms using images of size 1002×1002. Namely, we have trained 5 architectures: U-Net [145] (with a Resnet 34 backbone), DeepLabV3+ [146] (with a Resnet 50 backbone), Mask RCNN [147] (with a Resnet 50 backbone), HRNet-Seg [148] (with

Size	Architecture	Precision	Recall	F1-score
224×224	ResNet-50	**100** (100-100)	96.14 (94.55–97.73)	98.03 (96.88–99.18)
	ResNet-101	**100** (100–100)	97.62 (96.36–98.88)	98.79 (97.89–99.69)
	EfficientNet-B3	99.38 (98.73–100)	95.84 (94.19–97.49)	97.58 (96.31–98.85)
	FBNet	**100** (100–100)	97.62 (96.36–98.88)	98.79 (97.89–99.69)
512×512	ResNet-50	99.39 (98.75-100)	97.03 (95.63–98.43)	98.19 (97.09–99.2*)
	ResNet-101	99.39 (98.75–100)	97.32 (95.99–98.65)	98.35 (97.30–99.40)
	EfficientNet-B3	99.38 (98.73–100)	95.54 (93.84–97.24)	97.42 (96.11–98.73)
	FBNet	98.80 (97.90–99.70)	98.51 (97.51–99.51)	98.66 (97.71–99.61)
Resizing	ResNet-50	**100** (100-100)	**99.70** (99.25–100)	**99.85** (99.53–100)
	ResNet-101	**100** (100-100)	97.92 (96.74–99.10)	98.95 (98.11–99.79)
	EfficientNet-B3	97.94 (96.77–99.11)	98.81 (97.92–99.70)	98.37 (97.33–99.41)
	FBNet	**100** (100–100)	98.51 (97.51–99.51)	99.25 (98.54–99.96)

Table 2.13: Performance (95% CI) for the test set obtained by each classification model. The best result is highlighted in bold face.

Model	Dice coefficient	Jaccard index
DeepLabV3+	**95.66** (93.40–97.92)	**91.68** (88.62–94.74)
HRNet-seg	95.31 (95.97–97.65)	91.05 (87.89–94.21)
Mask-RCNN	91.18 (88.04–94.32)	83.80 (79.72–87.88)
U-Net	60.14 (54.72–65.56)	43.00 (37.52–48.48)
U^2-Net	66.94 (61.73–72.15)	50.31 (44.77–55.85)

Table 2.14: Performance (95% CI) for the test set obtained by each segmentation model. The best result is highlighted in bold face.

an HRNet W30 backbone) and U^2-net [149] (with its underlying backbone). As can be seen in Table 2.14, the best model is obtained using the DeepLabV3+ architecture with a Dice coefficient of 95.66% and a Jaccard index of 91.68%. A similar result is also obtained using the HRNet-seg architecture, whereas the other models obtain much worse results.

In order to annotate enough images to train our Deep Learning models, we presented a pipeline to analyse motility images using image processing techniques. Using this pipeline we produced a semi-automatic annotation of images that was validated and corrected by experts; hence, the tedious task of annotating images was simplified. This annotation method can be seen as a semi-supervised approach in which human beings intervene. Using this approach, we have created the first publicly available annotated dataset for measuring bacterial spread on motility images. Finally, we deployed our models in an open-source and user-friendly application called MotilityJ. This application internally makes a connection between the developed PyTorch models and ImageJ using the approach developed in the previous work of DeepClas4Bio.

MotilityJ was, at least up to the best our knowledge, the first application for the automatic analysis of bacteria spread in motility images. The underlying algorithms of MotilityJ are based on highly accurate Deep Learning models that generate a segmen-

tation comparable to those produced by experts; but, considerably reducing the effort required to obtain them. Thanks to the development of the Deep Learning models and their deployment in MotilityJ, the analysis of motility images will be faster, less subjective, more reliable and comparable among different laboratories all over the world. The developed tools will help to advance our understanding of the behaviour and virulence of bacteria.

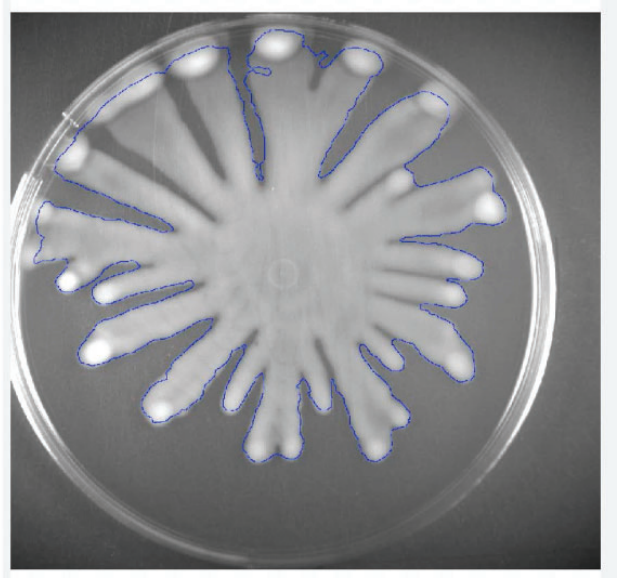


Figure 2.6: Example of the result obtained when segmenting a motility image to determine the spread of a bacteria that do not fully cover the Petri dishes with our tool.

2.5.2 Predicting Epitretinal Membrane

We focus now on the problem of diagnosis Epiretinal Membrane (ERM) from retinal fundus images. An epiretinal membrane is an eye disease that can lead to visual distortion and, in some cases, to loss of vision [150]. Screening retinal fundus images allows ophthalmologists to early detect and diagnose this disease; however, the manual interpretation of images is a time-consuming task. In spite of the existence of several computer vision tools for analysing retinal fundus images, they are mainly focused on the diagnosis of diabetic retinopathy and glaucoma. In collaboration with Hospital Vall D' Hebron, we conducted a thorough study of several Deep Learning architectures, and a variety of techniques to train them, in order to build a model for automatically diagnosing epiretinal membrane from retinal fundus images. In particular, we used 3 manually designed convolutional neural networks (namely, ResNet [151], ResNeSt [137] and HRnet [148]), 2 architectures found by neural architecture search (EfficientNet [152] and NasNet [77]); and 2 transformer-based architectures that are ViT [153], and its training efficient version, Deit [154].

In such a study, we tested several approaches for training Deep Learning architectures, including data augmentation using Generative Adversarial Networks, transfer learning from two different datasets, some bag of tricks (a set of techniques used in the literature to train image classification models and improve their performance), or ensemble methods. First of all, and in order to establish a baseline for our models, we used the transfer-learning method presented in [155] and used in ATLASS. The second approach consisted in using a Generative Adversarial Network (GAN) to synthesise new retinal images [156]. In particular, we trained a CycleGAN model [157] that allowed us to synthesise ERM images from healthy images and viceversa (1652 healthy images, and 1622 ERM images were generated using this procedure). The generated images were combined with the original dataset and used for training the new models. In the third set of experiments, we employed a bag of tricks that have been successfully employed in the literature to improve the performance of deep classification models. First of all, we replaced the Adam optimisation algorithm, the by-default optimiser used in FastAI, with the Ranger algorithm, which combines ideas from the RAdam optimisation algorithm [158] and the Lookahead optimiser [159]. Moreover, we used two regularisation techniques that are Label Smoothing [160] and MixUp [161]. Finally, we applied the cyclical learning rate policy for convergence proposed in [162]. In order to identify the benefits provided by each trick, an ablation study was conducted. From the ablation study, see Table 2.15, we can notice that there is not a single technique, or combination of techniques, that always produce the best results. However, the usage of Label Smoothing and MixUp as regularisation techniques consistently produced good results. It is also worth mentioning that the benefits obtained with each individual technique did not stack when combined with other techniques. This hinders the applicability of this bag of tricks since lots of experiments must be conducted to find which methods should be applied to produce the best result for each architecture.

The last approach that we explored to train our models was based on the fact that transfer learning produces better results when there is a close relation between the source and target task. Hence, we started by training the models with the RIADD dataset [163] (a dataset of 8289 images for multi-disease detection on retinal images); and, subsequently, we fine-tuned the models for our ERM dataset.

Finally, and in order to further improve the performance of our models, we employed ensemble methods. Namely, we tested the ensemble of several models [164], the application of test-time augmentation [165], and the combination of these two techniques.

The results presented in Table 2.16 show that the best approach to tackle the diagnosis of ERM consisted in ensembling a variety of models (both convolutional models and transformer-based architectures) that were pre-trained on a multi-disease detection dataset for fundus images, and then fine-tuned in our ERM dataset. Such an approach achieved a F1-score of 86.82%. This result highlights the advantages of close-transfer, that is, transfer learning from a close domain that allows us to take advantage of similar characteristics. In addition, it should be noted that the tools previously developed, see Sections 2.2 and 2.4, allowed us to perform this close-transfer in a simple way. Finally, we demonstrated the usage of occlusion based attribution (a technique that allowed us to estimate which areas of the image were critical for the classifiers' decision by occluding them and quantifying how the decision changed) to interpret the outputs

	B	R	F	L	M	RF	RM	FM	RL	FL	LM	RFL	RFM	RFLM
R-34	55.22	59.54	72.21	59.92	66.02	66.96	58.27	0.00	57.04	71.30	65.33	60.02	54.29	61.03
R-50	49.53	60.86	1.29	75.23	74.55	52.83	59.57	72.21	48.59	11.62	74.09	59.85	67.97	65.35
R-101	53.04	57.30	49.82	68.10	54.48	54.42	51.42	33.06	50.82	0.00	71.85	67.35	43.18	45.60
RS-26	55.18	58.54	0.42	66.67	64.85	72.43	62.60	72.21	57.52	72.21	71.48	73.62	60.92	72.17
RS-50	56.02	62.01	71.81	76.72	73.59	54.81	47.53	72.21	57.65	0.00	74.19	60.40	65.61	59.68
RS-50d	56.12	61.67	1.30	75.81	78.36	65.10	67.13	0.00	65.44	72.19	69.90	64.06	69.31	70.49
RS-101	59.03	60.64	39.11	74.35	76.31	58.37	49.34	2.98	59.93	7.49	70.49	59.68	68.33	62.07
E-0	51.16	55.89	0.00	72.10	69.60	68.81	62.65	68.25	55.60	25.54	67.27	66.75	73.83	73.83
E-1	48.62	55.84	56.43	69.03	67.28	66.67	55.13	0.00	63.96	0.00	68.41	70.14	66.49	66.58
E-2	60.20	63.49	71.98	67.42	66.92	69.21	55.93	58.75	58.69	0.00	65.82	71.14	70.49	65.59
E-3	56.68	61.20	71.22	72.81	57.55	68.91	57.52	69.22	60.09	64.04	67.88	73.41	64.65	68.26
ViT-244	69.41	68.40	0.00	72.21	70.92	0.00	67.20	72.21	72.21	72.21	72.21	0.00	72.21	0.00
ViT-384	81.29	67.39	67.74	60.84	59.74	65.14	66.28	2.17	38.07	62.15	57.28	66.21	66.13	66.27
Deit-384	74.85	70.73	72.21	80.39	81.52	63.86	76.94	72.21	72.21	72.18	77.77	72.66	77.51	76.40
N-050	55.55	59.72	27.19	58.63	50.18	54.38	52.55	71.30	54.45	70.60	54.17	52.73	52.22	52.96
H-32	73.74	55.23	77.51	77.51	80.50	79.82	53.81	78.27	76.74	70.34	74.27	79.26	54.30	50.75
H-40	71.09	54.00	58.89	69.76	70.76	58.93	60.99	24.38	61.92	70.11	65.04	64.56	66.59	62.06
H-44	72.33	55.33	62.57	63.85	69.96	67.32	66.16	70.28	71.30	64.17	70.46	69.59	68.29	67.33
H-48	70.60	49.42	70.53	73.95	73.49	72.34	73.90	67.68	67.74	67.69	69.58	72.26	73.55	72.86
H-64	73.78	55.02	71.24	76.56	77.88	61.89	72.01	0.00	0.85	69.60	70.34	66.75	72.70	74.47

F1-score colour scale: 0 | 18 | 36 | 54 | 72 | 90

Table 2.15: Ablation study of the bag of tricks using F1-score as metric. Each column represents a technique (B: baseline, R: Ranger optimiser, F: Flat cosine annealing, L: Label Smoothing, M: MixUp; and the rest of columns are combinations of the previous techniques). Each row represents an architecture (R: Resnet, RS: ResNeSt, E: EfficientNet, ViT: ViT, Deit: Deit, N: Nasnet, H: HRNet).

produced by our models, see Figure 2.7.

In this work, we thoroughly studied several approaches to build deep learning models for diagnosing epiretinal membrane. The best results, with a F1-score of 86.82%, was achieved by using the HRNet and transformer-based architectures, and combining 3 techniques (transfer learning from the RIADD dataset, test-time augmentation and model ensemble).

In summary, the seventh contribution of this work is the resolution of two real problems applying the techniques and knowledge developed during this work. First, we have developed an open-source application, called MotilityJ, able to automatically analyse the bacteria spread in motility images. Thanks to this work, the analysis of motility images will be faster, less subjective, more reliable and comparable among different laboratories all over the world. Second, we have build deep learning models for diagnosing epiretinal membrane surpassing the results obtained previously. Thanks to this work, we have a solution capable of assisting in the detection of epiretinal membrane in fundus images.

Architecture	Baseline	CycleGAN	Tricks	Transfer	TTA	RIADD
Resnet-34	55.22	55.18	72.21	59.09	65.69	75.04
Resnet-50	49.53	58.04	75.23	72.18	72.63	73.93
Resnet-101	53.04	46.53	71.85	72.20	72.20	68.38
Resnest-26	55.18	53.59	73.62	62.68	66.36	75.57
Resnest-50	56.02	56.22	76.72	49.22	55.57	75.76
Resnest50d_4s2x40d	56.12	61.99	78.36	63.38	68.10	73.05
Resnest101	59.03	49.63	76.31	56.92	64.00	76.07
EfficientNet-B0	51.16	60.47	73.83	67.43	65.05	78.87
EfficientNet-B1	48.62	47.26	70.14	66.09	71.16	79.05
EfficientNet-B2	60.20	49.94	71.98	61.82	65.30	79.19
EfficientNet-B3	56.68	50.20	73.41	66.96	65.67	79.45
VIT-B/16-244	69.41	62.80	72.21	73.13	76.25	83.01
ViT-B/16-R50-384	*81.29*	62.91	67.39	83.86	84.23	87.44
Deit-B/16-384	74.85	72.11	*81.52*	76.46	76.77	87.01
Nasnet-050	55.55	49.76	71.30	55.23	50.66	52.65
HRNet-w32	73.74	67.15	80.50	79.22	81.17	*87.98*
HRNet-w40	71.09	52.53	70.76	***84.00***	*85.52*	87.30
HRNet-w44	72.33	60.79	71.30	82.61	83.27	87.50
HRNet-w48	70.60	*76.12*	73.95	82.17	84.59	86.32
HRNet-w64	73.78	50.27	77.88	83.70	84.35	87.59

Table 2.16: F1-score achieved by the studied architectures using the baseline procedure, the CycleGAN dataset, the bag-of-tricks, and transfer learning from a close domain. Moreover, we include the results obtained by applying test-time augmentation (TTA) to the models fine-tuned from a close domain, and the results for the RIADD dataset. In italics the best model for each approach, and in bold face the best overall model without TTA.

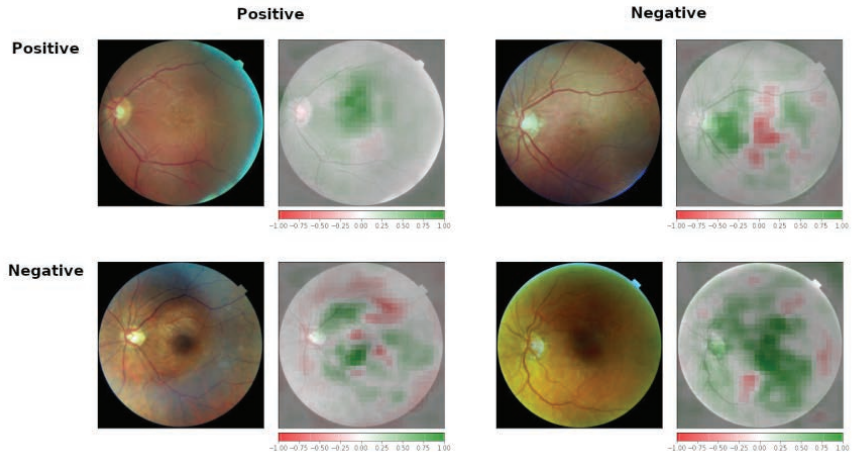


Figure 2.7: Sample of occlusion-based attribution confusion matrix on the HRNet-w40 model. The red pixels in the heatmap indicate a negative attribution region (areas whose absence increases the score), whereas the green pixels indicate a positive attribution region (areas whose presence increase the prediction score).

Chapter 3

Contributions

In this chapter, the main contributions that make up the memoir are presented. In particular, for each of these contributions its summary and a small table with its most important data is presented.

3.1 DeepClas4Bio: Connecting bioimaging tools with Deep Learning frameworks for image classification

Deep Learning techniques have been successfully applied to tackle several image classification problems in bioimaging. However, the models created from Deep Learning frameworks cannot be easily accessed from bioimaging tools such as ImageJ or Icy; this means that life scientists are not able to take advantage of the results obtained with those models from their usual tools. In this paper, we aim to facilitate the interoperability of bioimaging tools with Deep Learning frameworks. In this project, called DeepClas4Bio, we have developed an extensible API that provides a common access point for classification models of several Deep Learning frameworks. In addition, this API might be employed to compare Deep Learning models, and to extend the functionality of bioimaging programs by creating plugins. Using the DeepClas4Bio API, we have developed a metagenerator to easily create ImageJ plugins. In addition, we have implemented a Java application that allows users to compare several Deep Learning models in a simple way using the DeepClas4Bio API. Moreover, we present three examples where we show how to work with different models and frameworks included in the DeepClas4Bio API using several bioimaging tools — namely, ImageJ, Icy and ImagePy. This project brings to the table benefits from several perspectives. Developers of Deep Learning models can disseminate those models using well-known tools widely employed by life-scientists. Developers of bioimaging programs can easily create plugins that use models from Deep Learning frameworks. Finally, users of bioimaging tools have access to powerful tools in a known environment for them.

DeepClas4Bio: Connecting bioimaging tools with Deep Learning frameworks for image classification

Authors	Adrián Inés, César Domínguez, Jónathan Heras, Eloy Mata and Vico Pascual
Journal	Computers in Biology and Medicine
Impact factor	3.434 (2019)
Rank	Q1 (JCR)
Publisher	Elsevier
Volume	108
Issue	-
Pages	49-56
Year	2019
Month	May
ISSN	0010-4825
DOI	10.1016/j.compbiomed.2019.03.026
URL	`https://www.sciencedirect.com/science/article/pii/S0010482519301052`
State	Published
Cites	10 (Google Scholar)
Project webpage	`https://github.com/adines/DeepClas4Bio`
Project stats	69.113 downloads (`https://pepy.tech/`)
Author's contribution	The PhD student is the main author of the paper. He has been in charge of all the development processes, research; analysis, design and implementation of the API and the experiments. In addition, he has written the manuscript, and was in charge of the reviewed process.

3.2 CLoDSA: A tool for augmentation in classification, localization, detection, semantic segmentation and instance segmentation tasks

Deep Learning techniques have been successfully applied to bioimaging problems; however, these methods are highly data demanding. An approach to deal with the lack of data and avoid overfitting is the application of data augmentation, a technique that generates new training samples from the original dataset by applying different kinds of transformations. Several tools exist to apply data augmentation in the context of image classification, but it does not exist a similar tool for the problems of localization, detection, semantic segmentation or instance segmentation that works not only with 2 dimensional images but also with multi-dimensional images (such as stacks or videos). In this paper, we present a generic strategy that can be applied to automatically augment a dataset of images, or multi-dimensional images, devoted to classification, localization, detection, semantic segmentation or instance segmentation. The augmentation method presented in this paper has been implemented in the open-source package CLoDSA. To prove the benefits of using CLoDSA, we have employed this library to improve the accuracy of models for Malaria parasite classification, stomata detection, and automatic segmentation of neural structures. CLoDSA is the first, at least up to the best of our knowledge, image augmentation library for object classification, localization, detection, semantic segmentation, and instance segmentation that works not only with 2 dimensional images but also with multi-dimensional images.

CLoDSA: A tool for augmentation in classification, localization, detection, semantic segmentation and instance segmentation tasks

Authors	Ángela Casado-García, César Domínguez, Manuel García-Domínguez, Jónathan Heras, Adrián Inés, Eloy Mata and Vico Pascual
Journal	BMC Bioinformatics
Impact factor	3.242 (2019)
Rank	Q1 (JCR)
Publisher	Springer Nature
Volume	20
Issue	-
Pages	-
Year	2019
Month	June
ISSN	1471-2105
DOI	10.1186/s12859-019-2931-1
URL	`https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-2931-1`
State	Published
Cites	38 (Google Scholar)
Project webpage	`https://github.com/joheras/CLoDSA`
Project stats	110.343 downloads (`https://pepy.tech/`)
Author's contribution	The PhD student was in charge of implementing some of the methods provided by CLoDSA, testing this library, and use it for building the models presented in the Results section. Also, he has read and approved the final manuscript.

3.3 Biomedical image classification made easier thanks to transfer and semi-supervised learning

Deep Learning techniques are the state-of-the-art approach to solve image classification problems in biomedicine; however, they require the acquisition and annotation of a considerable volume of images. In addition, using Deep Learning libraries and tuning the hyperparameters of the networks trained with them might be challenging for several users. These drawbacks prevent the adoption of these techniques outside the machine-learning community. In this work, we present an Automated Machine Learning (AutoML) method to deal with these problems. Our AutoML method combines transfer learning with a new semi-supervised learning procedure to train models when few annotated images are available. In order to facilitate the dissemination of our method, we have implemented it as an open-source tool called ATLASS. Finally, we have evaluated our method with two benchmarks of biomedical image classification datasets. Our method has been thoroughly tested both with small datasets and partially annotated biomedical datasets; and, it outperforms, both in terms of speed and accuracy, the existing AutoML tools when working with small datasets; and, might improve the accuracy of models up to a 10% when working with partially annotated datasets. The work presented in this paper allows the use of Deep Learning techniques to solve an image classification problem with few resources. Namely, it is possible to train deep models with small, and partially annotated datasets of images. In addition, we have proven that our AutoML method outperforms other AutoML tools both in terms of accuracy and speed when working with small datasets.

Biomedical image classification made easier thanks to transfer and semi-supervised learning

Authors	Adrián Inés, César Domínguez, Jónathan Heras, Eloy Mata and Vico Pascual
Journal	Computer Methods and Programs in Biomedicine
Impact factor	5.428 (2020)
Rank	Q1 (JCR)
Publisher	Elsevier
Volume	198
Issue	-
Pages	105782
Year	2021
Month	January
ISSN	0169-2607
DOI	10.1016/j.cmpb.2020.105782
URL	`https://www.sciencedirect.com/science/article/pii/S0169260720316151`
State	Published
Cites	16 (Google Scholar)
Project webpage	`https://github.com/adines/ATLASS`
Project stats	-
Author's contribution	The PhD student is the main author of the paper. He has been in charge of all the development process, research, analysis, design and implementation of the method, the API and the experiments. In addition, he has written the manuscript, and was in charge of the reviewed process.

3.4 MotilityJ: An open-source tool for the classification and segmentation of bacteria on motility images

Infectious diseases produced by antimicrobial resistant microorganisms are a major threat to human, and animal health worldwide. This problem is increased by the virulence and spread of these bacteria. Surface motility has been regarded as a pathogenicity element because it is essential for many biological functions, but also for disease spreading; hence, investigations on the motility behaviour of bacteria are crucial to understand chemotaxis, biofilm formation and virulence in general. To identify a motile strain in the laboratory, the bacterial spread area is observed on media solidified with agar. Up to now, the task of measuring bacteria spread was a manual, and, therefore, tedious and time-consuming task. The aim of this work is the development of a set of tools for bacteria segmentation in motility images. In this work, we address the problem of measuring bacteria spread on motility images by creating an automatic pipeline based on Deep Learning models. Such a pipeline consists of a classification model to determine whether the bacteria has spread to cover completely the Petri dish, and a segmentation model to determine the spread of those bacteria that do not fully cover the Petri dishes. In order to annotate enough images to train our Deep Learning models, a semi-automatic annotation procedure is presented. The classification model of our pipeline achieved a F1-score of 99.85%, and the segmentation model achieved a Dice coefficient of 95.66%. In addition, the segmentation model produces results that are indistinguishable, and in many cases preferred, from those produced manually by experts. Finally, we facilitate the dissemination of our pipeline with the development of MotilityJ, an open-source and user-friendly application for measuring bacteria spread on motility images. In this work, we have developed an algorithm and trained several models for measuring bacteria spread on motility images. Thanks to this work, the analysis of motility images will be faster and more reliable. The developed tools will help to advance our understanding of the behaviour and virulence of bacteria.

MotilityJ: An open-source tool for the classification and segmentation of bacteria on motility images

Authors	Ángela Casado-García, Gabriela Chichón, César Domínguez, Manuel García-Domínguez, Jónathan Heras, Adrián Inés, María López, Eloy Mata, Vico Pascual and Yolanda Sáenz
Journal	Computers in Biology and Medicine
Impact factor	4.589 (2020)
Rank	Q1 (JCR)
Publisher	Elsevier
Volume	136
Issue	-
Pages	104673
Year	2021
Month	September
ISSN	0010-4825
DOI	10.1016/j.compbiomed.2021.104673
URL	`https://www.sciencedirect.com/science/article/pii/S0010482521004674`
State	Published
Cites	4 (Google Scholar)
Project webpage	`https://github.com/joheras/MotilityJ`
Project stats	12.569 downloads (`https://pepy.tech/`)
Author's contribution	The PhD student was in charge of designing, developing and testing the classification models implemented int MotilityJ. Also, he has read and approved the final manuscript.

3.5 Prediction of Epiretinal Membrane from Retinal Fundus Images Using Deep Learning

An epiretinal membrane (ERM) is an eye disease that can lead to visual distortion and, in some cases, to loss of vision. Screening retinal fundus images allows ophthalmologists to early detect and diagnose this disease; however, the manual interpretation of images is a time-consuming task. In spite of the existence of several computer vision tools for analysing retinal fundus images, they are mainly focused on the diagnosis of diabetic retinopathy and glaucoma. In this work, we have conducted a thorough study of several Deep Learning architectures, and a variety of techniques to train them, in order to build a model for automatically diagnosing ERM. As a result, we have built several models that can be ensembled to achieve a F1-score of 86.82%. The lessons learned in this work can serve as a basis for the construction of Deep Learning models for diagnosing other eye diseases.

Prediction of Epiretinal Membrane from Retinal Fundus Images Using Deep Learning

Authors	Ángela Casado-García, Manuel García-Domínguez, Jónathan Heras, Adrián Inés, Didac Royo, Miguel Ángel Zapata
Journal	Lecture Notes in Computer Science
Impact factor	0.25 (SJCR)
Rank	Q3 (SJCR)
Publisher	Springer
Volume	12882
Issue	-
Pages	3-13
Year	2021
Month	September
ISSN	0302-9743
DOI	10.1007/978-3-030-85713-4_1
URL	https://link.springer.com/chapter/10.1007/978-3-030-85713-4_1
State	Published
Cites	-
Project webpage	https://github.com/CoVUR/ERM
Project stats	-
Author's contribution	The PhD student, was in charge of designing, developing and testing the bag of tricks approach of the experiments. Also, he has read and approved the final manuscript.

3.6 Semi-Supervised Learning for Image Classification using Compact Networks in the BioMedical Context

The development of mobile and on the edge applications that embed deep convolutional neural models has the potential to revolutionise biomedicine. However, most Deep Learning models require computational resources that are not available in smartphones or edge devices; an issue that can be faced by means of compact models that require less resources than standard Deep Learning models. The problem with such models is that they are, at least usually, less accurate than bigger models. In this work, we study how this limitation can be addressed with the application of semi-supervised learning techniques. We conduct several statistical analyses to compare performance of deep compact architectures when trained using semi-supervised learning methods for tackling image classification tasks in the biomedical context. In particular, we explore three families of compact networks (manually designed, automatically designed, and quantized), and two families of semi-supervised learning techniques (self-training and consistency regularisation methods) for 10 biomedical tasks. By combining semi-supervised learning methods with compact networks such as FBNet, MixNet, MNasNet and ResNet-18, it is possible to obtain a similar performance to standard size networks. In general, the best results are obtained when combining data distillation (a self-training method) with MixNet, and plain distillation (another self-training method) with ResNet-18. Another conclusion that we can draw from our study is that, in general, NAS networks obtain better results than manually designed networks and quantized networks, with the exception of ResNet-18 that obtains similar results than NAS networks. Regarding to semi-supervised learning methods, the data & model distillation method is the best option for standard size networks; data distillation produces the best results for NAS networks; and for the other types of networks, there is no a general rule, although the data distillation approach generally produces good results. Finally, we have developed a Python library in order to facilitate the combination of compact networks and semi-supervised learning methods to tackle image classification tasks. The work presented in this paper shows the benefits of apply semi-supervised methods to compact networks. Namely, combining semi-supervised methods and compact networks, we can create compact models that are not only as accurate as standard size models, but also faster and lighter. Finally, to facilitate the application of the methods studied in this work, we have developed a library that simplifies the construction of compact models using semi-supervised learning methods.

Semi-Supervised Learning for Image Classification using Compact Networks in the BioMedical Context

Authors	Adrián Inés, Andrés Díaz-Pinto, César Domínguez, Jónathan Heras, Eloy Mata and Vico Pascual
Journal	arXiv
Impact factor	-
Rank	-
Publisher	Cornell University
Volume	-
Issue	-
Pages	-
Year	2022
Month	May
ISSN	2331-8422
DOI	10.48550/ARXIV.2205.09678
URL	`https://arxiv.org/abs/2205.09678`
State	Preprint
Cites	-
Project webpage	`https://github.com/adines/SemiCompact`
Project stats	20.146 downloads (`https://pepy.tech/`)
Author's contribution	The PhD student is the main author of the paper. He has been in charge of all the development process, research, analysis, design and implementation of the method, the API and the experiments. In addition, he has written the manuscript, and was in charge of the reviewed process.

Chapter 4

Conclusions and further work

In this work, we have addressed the problem of democratising the use of Deep Learning techniques to solve image classification problems. Specifically, we have developed different tools and techniques that try to solve three challenges in the use of Deep Learning techniques for image classification that are: the reduction of the amount of data necessary to use these techniques, the democratisation of the construction of Deep Learning models, and the democratisation of the use of Deep Learning models.

To reduce the amount of necessary data to use Deep Learning techniques, we studied two different approaches. In the former, we used a technique that consists in augmenting the data by creating new artificial data. For this, we developed a data augmentation framework called CLoDSA, that allows us to increase the amount of images for a wide variety of computer vision problems by using different data augmentation techniques. This tool allows us to perform data augmentation not only on classification problems, but also on detection and segmentation problems. In the future, we plan to expand the functionality of CLoDSA by including more augmentation techniques, integrating more frameworks and including GAN based methods that provide new data augmentation.

In addition, we studied the use of unlabelled data to train Deep Learning models for image classification by creating new semi-supervised methods. Specifically, we developed two semi-supervised methods. The first one is based on the concept of data and model distillation that has allowed us to increase the performance of Deep Learning models by 10% using a reduced number of annotated images. In the future, we want to be able to automatically select the transformations and models to be used in this semi-supervised method depending on the type of problem to be solved. In addition, we want to transfer this method to other computer vision tasks, such as, object detection or image segmentation. We also developed a semi-supervised method based on the topological data analysis techniques that improves up to 16% the classical annotation methods. In the future, it is intended to extend this method to no binary classification problems, as well as the investigation of new TDA techniques.

Furthermore, we are studying the few-shot learning problem, in collaboration with the research group of Professor Thomas Brox of the Albert-Ludwigs University of Freiburg. The few-shot learning problem is an image classification problem where

the training dataset contains limited information; that is, a limited number of labelled images per class. In this collaboration, we are working on an approach based on Generative Adversarial Networks.

The second challenge that we addressed was the democratisation of the construction of Deep Learning models. To solve this problem, we developed an AutoML tool, called ATLASS, that assists the user in the entire process of creating a classification Deep Learning model, from annotating the images to creating the Deep Learning model. In this framework, one of the semi-supervised methods developed before was used in order to improve the accuracy of the models obtained as well as give the possibility of using unlabelled images. The developed framework can be used both on a local server, with the complexity of maintenance and configuration that this entails, or can be used in the cloud, with Google Colaboratoy or Amazon Web Services for example, with all the security problems that this entails; hence, in the future we want to improve the security of our Deep Learning models by using data privacy techniques.

To address the third challenge, the democratisation of the use of Deep Learning models, we studied two approaches. In the former, we tried to reduce the amount of computational resources necessary for the use of Deep Learning models, as well as reduce their size. For this, we carried out an exhaustive study of the combination of compact networks and quantisation techniques with semi-supervised learning methods. As a result, we were able to conclude that by combining these two techniques we can obtain compact models that are as accurate as standard-size models and also faster and lighter. This fact allows us to include these models in smartphones or edge devices. Our next goal is to transfer this study to other computer vision problems trying to replace existing backbones with compact networks. Moreover, we developed a library called DeepClas4Bio that groups the main Deep Learning frameworks and facilitates the interoperability between these frameworks and the main bioimaging tools, such as ImageJ or Icy. In addition, we developed some plugins for each of these tools that allow an easy integration with DeepClas4Bio. We also intend to extend our framework to other computer vision tasks, such as object detection or image segmentation.

Finally, the techniques and methods developed in this work were used in two real biomedical problems, that were the measurement of the propagation of bacteria in motility images, and the detection of the epiretinal membrane disease from fundus images. This allowed us to see that the developed methods have an impact on real life problems, a research line that we will keep exploring in the future.

Chapter 5

Conclusiones y trabajo futuro

En este trabajo hemos abordado el problema de democratizar el uso de las técnicas de *Deep Learning* para resolver problemas de clasificación de imágenes. En concreto, hemos desarrollado diferentes herramientas y técnicas que intentan solucionar tres retos en el uso de técnicas de *Deep Learning* que son: la reducción de la cantidad de datos necesarios para utilizar estas técnicas, la democratización de la construcción de modelos de *Deep Learning*, y la democratización del uso de modelos de *Deep Learning*.

Para reducir la cantidad de datos necesarios para utilizar técnicas de *Deep Learning*, hemos estudiado dos enfoques diferentes. En el primero hemos utilizado una técnica que consiste en aumentar los datos creando nuevos datos artificiales. Para ello, hemos desarrollado un *framework* de aumento de datos llamado CLoDSA, que nos permite aumentar la cantidad de imágenes de un problema utilizando diferentes técnicas de aumento de datos. Esta herramienta nos permite realizar aumento de datos no solo en problemas de clasificación, sino también en otros problemas de visión por computador. En el futuro, planeamos expandir la funcionalidad de CLoDSA al incluir más técnicas de aumento, integrar más *frameworks* e incluir métodos basados en GAN (*Generative Adversarial Networks*) que permitan el aumento de nuevos datos.

Además, hemos estudiado el uso de datos no etiquetados mediante la creación de nuevos métodos semi-supervisados. En concreto, hemos desarrollado dos métodos semi-supervisados. El primero se basa en el concepto de destilación de datos y modelos que nos ha permitido aumentar el rendimiento de los modelos de *Deep Learning* en un 10% utilizando un número reducido de imágenes anotadas. En el futuro queremos poder seleccionar automáticamente las transformaciones y modelos a utilizar de los métodos semi-supervisados dependiendo del tipo de problema a resolver. Además, queremos trasladar estos métodos a otras tareas de visión artificial, como la detección de objetos o la segmentación de imágenes. También hemos desarrollado un método semi-supervisado basado en las técnicas de análisis de datos topológicos que mejora hasta un 16% los métodos clásicos de anotación. En el futuro se pretende extender este método a problemas de clasificación no binaria, así como a la investigación de nuevas técnicas de TDA.

Además, estamos estudiando el problema del *few-shot learning*, en colaboración

con el grupo de investigación del profesor Thomas Brox de la Universidad Albert-Ludwigs de Friburgo. El problema de *few-shot learning* es un problema de clasificación de imágenes en el que el conjunto de datos de entrenamiento contiene información limitada, es decir, pocas imágenes etiquetadas por clase. En esta colaboración estamos trabajando en un enfoque de redes generativas.

El segundo reto que hemos abordado es la democratización de la construcción de modelos de *Deep Learning*. Para resolver este problema, hemos desarrollado una herramienta de AutoML, llamada ATLASS, que ayuda al usuario en todo el proceso de creación de un modelo de clasificación de aprendizaje profundo, desde la anotación de las imágenes hasta la creación del modelo de aprendizaje profundo. En este marco, se ha utilizado uno de los métodos semi-supervisados desarrollados anteriormente para mejorar la precisión de los modelos obtenidos así como dar la posibilidad de utilizar imágenes sin etiquetar. El *framework* desarrollado se puede utilizar tanto en un servidor local, con la complejidad de mantenimiento y configuración que ello conlleva, o se puede utilizar en la nube, con Google Colaboratoy o Amazon Web Services por ejemplo, con todos los problemas de seguridad que ello conlleva; es por ello que en el futuro queremos mejorar la seguridad de nuestros modelos de aprendizaje profundo mediante el uso de técnicas de privacidad de datos.

Para abordar el tercer desafío, la democratización del uso de modelos de *Deep Learning*, hemos estudiado dos enfoques. En el primero, se ha intentado reducir la cantidad de recursos computacionales necesarios para el uso de modelos de *Deep Learning*, así como reducir su tamaño. Para ello, hemos realizado un estudio exhaustivo de la combinación de redes compactas y técnicas de cuantificación con métodos de aprendizaje semi-supervisado. Como resultado, hemos podido concluir que combinando estas dos técnicas podemos obtener modelos compactos, tan precisos como los modelos de tamaño estándar y también más rápidos y ligeros. Este hecho nos permite incluir estos modelos en *smartphones* o dispositivos *edge*. Nuestro siguiente objetivo es trasladar este estudio a la segmentación de imágenes, intentando sustituir los *backbones* existentes por redes compactas. Además, hemos desarrollado una librería llamada DeepClas4Bio que agrupa los principales *frameworks* de *Deep Learning* y facilita la interoperabilidad entre estos *frameworks* y las principales herramientas de bioimagen, como ImageJ o Icy. Además, se han desarrollado unos *plugins* para cada una de estas herramientas que permiten una fácil integración con DeepClas4Bio. También tenemos la intención de extender nuestro *framework* a otras tareas de visión artificial, como la detección de objetos o la segmentación de imágenes.

Finalmente, las técnicas y métodos desarrollados en este trabajo se han utilizado en dos problemas biomédicos reales, que son la medición de la propagación de bacterias en imágenes de motilidad, y la detección de la enfermedad de la membrana epirretiniana a partir de imágenes de fondo de ojo. Esto nos ha permitido ver que los métodos desarrollados tienen impacto en problemas de la vida real, línea de investigación que seguiremos explorando en el futuro.

References

[1] James Moor. "The Dartmouth College artificial intelligence conference: The next fifty years". In: *Ai Magazine* 27.4 (2006), pp. 87–87. DOI: 10.1609/aimag.v27i4.1911.

[2] Viktor Mayer-Schönberger and Kenneth Cukier. *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt, 2013.

[3] Saeed H Alsamhi, Ou Ma, and Mohd Samar Ansari. "Survey on artificial intelligence based techniques for emerging robotic communication". In: *Telecommunication Systems* 72.3 (2019), pp. 483–503. DOI: 10.1007/s11235-019-00561-z.

[4] Carlos Affonso et al. "Deep learning for biological image classification". In: *Expert Systems with Applications* 85 (2017), pp. 114–122. DOI: 10.1016/j.eswa.2017.05.039.

[5] Teresa Araújo and et al. "Classification of breast cancer histology images using convolutional neural networks". In: *PloS one* 12.6 (2017), e0177544. DOI: 10.1371/journal.pone.0177544.

[6] Kimon Kieslich, Marco Lünich, and Frank Marcinkowski. "The Threats of Artificial Intelligence Scale (TAI)". In: *International Journal of Social Robotics* (2021), pp. 1–15. DOI: 10.1007/s12369-020-00734-w.

[7] Bertrand Braunschweig and Malik Ghallab. *Reflections on Artificial Intelligence for Humanity*. Springer, 2021.

[8] KP Smith and JE Kirby. "Image analysis and artificial intelligence in infectious disease diagnostics". In: *Clinical Microbiology and Infection* 26.10 (2020), pp. 1318–1323. DOI: 10.1016/j.cmi.2020.03.012.

[9] Shervin Emami and Valentin Petrut Suciu. "Facial recognition using OpenCV". In: *Journal of Mobile, Embedded and Distributed Systems* 4.1 (2012), pp. 38–43.

[10] Joel Janai et al. "Computer vision for autonomous vehicles: Problems, datasets and state of the art". In: *Foundations and Trends® in Computer Graphics and Vision* 12.1–3 (2020), pp. 1–308. DOI: 10.1561/0600000079.

[11] Jonathon S Hare et al. “Mind the Gap: Another look at the problem of the semantic gap in image retrieval”. In: *Proceedings of the Multimedia Content Analysis, Management, and Retrieval*. Vol. 6073. 2006, p. 607309. DOI: 10.1117/12.647755.

[12] Guillermo Marqués, Thomas Pengo, and Mark A. Sanders. “Science Forum: Imaging methods are vastly underreported in biomedical research”. In: *eLife 9:e55133* (2020). DOI: 10.7554/eLife.55133.

[13] Erik Meijering. “A bird’s-eye view of deep learning in bioimage analysis”. In: *Computational and Structural Biotechnology Journal* 18 (2020), pp. 2312–2325. DOI: 10.1016/j.csbj.2020.08.003.

[14] Vijai Singh, Namita Sharma, and Shikha Singh. “A review of imaging techniques for plant disease detection”. In: *Artificial Intelligence in Agriculture* 4 (2020), pp. 229–242. DOI: 10.1016/j.aiia.2020.10.002.

[15] Christian Scheeder, Florian Heigwer, and Michael Boutros. “Machine learning and image-based profiling in drug discovery”. In: *Current opinion in systems biology* 10 (2018), pp. 43–52. DOI: 10.1016/j.coisb.2018.05.004.

[16] Manuel Campos-Taberner et al. “Understanding deep learning in land use classification based on Sentinel-2 time series”. In: *Scientific reports* 10.1 (2020), pp. 1–12. DOI: 10.1038/s41598-020-74215-5.

[17] Ali Madani, and others. “Fast and accurate view classification of echocardiograms using deep learning”. In: *Nature Partner Journals Digital Medicine* 1.6 (2018). DOI: 10.1038/s41746-017-0013-1.

[18] Korsuk Sirinukunwattana, et al. “Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images”. In: *IEEE Transactions on Medical Imaging* 35.5 (2016), pp. 1196–1206. DOI: 10.1109/TMI.2016.2525803.

[19] Kai-Lung Hua, et al. “Computer-aided classification of lung nodules on computed tomography images via deep learning technique”. In: *OncoTargets and Therapy* 8 (2015), pp. 2015–2022. DOI: 10.2147/OTT.S80733.

[20] Andre Esteva, et al. “Dermatologist-level classification of skin cancer with deep neural networks”. In: *Nature* 542 (2017), pp. 115–118. DOI: 10.1038/nature21056.

[21] Florentino Luciano Caetano dos Santos et al. “Computer vision for virus image classification”. In: *Biosystems Engineering* 138 (2015), pp. 11–22. DOI: 10.1016/j.biosystemseng.2015.01.005.

[22] José M Peña et al. “Object-based image classification of summer crops with machine learning methods”. In: *Remote Sensing* 6.6 (2014), pp. 5019–5041. DOI: 10.3390/rs6065019.

[23] Anna Bosch, Andrew Zisserman, and Xavier Munoz. “Image classification using random forests and ferns”. In: *Proceedings of the IEEE 11th international conference on computer vision (ICCV’07)*. IEEE. 2007, pp. 1–8. DOI: 10.1109/ICCV.2007.4409066.

[24] Issam El Naqa and Martin J Murphy. "What is machine learning?" In: *machine learning in radiation oncology*. Springer, 2015, pp. 3–11. DOI: 10.1007/978-3-319-18305-3_1.

[25] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006. ISBN: 0387310738.

[26] Timo Ojala, Matti Pietikainen, and David Harwood. "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions". In: *Proceedings of 12th international conference on pattern recognition (ICPR'94)*. Vol. 1. IEEE. 1994, pp. 582–585. DOI: 10.1109/ICPR.1994.576366.

[27] David G Lowe. "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2 (2004), pp. 91–110. DOI: 10.1023/B:VISI.0000029664.99615.94.

[28] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features". In: *Proceedings of the European conference on computer vision (ECCV'06)*. Springer. 2006, pp. 404–417. DOI: 10.1007/11744023_3.

[29] Ethan Rublee et al. "ORB: An efficient alternative to SIFT or SURF". In: *Proceedings of the International conference on computer vision (ICCV'11)*. Ieee. 2011, pp. 2564–2571. DOI: 10.1109/ICCV.2011.6126544.

[30] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444. DOI: 10.1038/nature14539.

[31] Frank Rosenblatt. *Principles of neurodynamics. Perceptrons and the theory of brain mechanisms*. Tech. rep. Cornell Aeronautical Lab Inc Buffalo NY, 1961.

[32] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network". In: *Proceedings of the International conference on engineering and technology (ICET'17)*. 2017, pp. 1–6.

[33] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks". In: *Proceedings of the International conference on machine learning (ICML'13)*. 2013, pp. 1310–1318.

[34] Jie Zhou et al. "Graph neural networks: A review of methods and applications". In: *AI Open* 1 (2020), pp. 57–81.

[35] Salman Khan et al. "Transformers in vision: A survey". In: *ACM Computing Surveys (CSUR)* (2021).

[36] Adrian Rosebrock. *Deep Learning for Computer Vision with Python*. PyImageSearch, 2017.

[37] Zahangir Alom et al. "The history began from alexnet: A comprehensive survey on deep learning approaches". In: *arXiv* (2018). DOI: arXiv:1803.01164.

[38] Chen Sun et al. "Revisiting unreasonable effectiveness of data in deep learning era". In: *Proceedings of the IEEE international conference on computer vision (ICCV'17)*. 2017, pp. 843–852. DOI: 10.1109/ICCV.2017.97.

[39] Jia Deng et al. “ImageNet: A large-scale hierarchical image database”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’09)*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[40] Tsung-Yi Lin et al. “Microsoft COCO: Common Objects in Context”. In: *European Conference on Computer Vision (ECCV’14)*. Ed. by David Fleet et al. 2014, pp. 740–755. DOI: 10.1007/978-3-319-10602-1_48.

[41] Jeremy Irvin et al. “Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison”. In: *Proceedings of the AAAI conference on artificial intelligence (AAAI’19)*. Vol. 33. 01. 2019, pp. 590–597. DOI: 10.1609/aaai.v33i01.3301590.

[42] Peng Sun et al. “Optimizing network performance for distributed dnn training on gpu clusters: Imagenet/alexnet training in 1.5 minutes”. In: *arXiv* (2019). DOI: arXiv:1902.06855.

[43] Ali Sharif Razavian et al. “CNN features off-the-shelf: an astounding baseline for recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops (ICCV’14)*. 2014, pp. 806–813. DOI: 10.1109/CVPRW.2014.131.

[44] A. Kumar et al. “An Ensemble of Fine-Tuned Convolutional Neural Networks for Medical Image Classification”. In: *IEEE Journal of Biomedical and Health Informatics* 21.1 (2017), pp. 31–40. DOI: 10.1109/JBHI.2016.2635663.

[45] Grant J. Scott et al. “Training Deep Convolutional Neural Networks for Land–Cover Classification of High-Resolution Imagery”. In: *IEEE Geoscience and Remote Sensing Letters* 14.4 (2017), pp. 549–553. DOI: 10.1109/LGRS.2017.2657778.

[46] Zar Nawab Khan Swati et al. “Brain tumor classification for MR images using transfer learning and fine-tuning”. In: *Computerized Medical Imaging and Graphics* 75 (2019), pp. 34–46. DOI: 10.1016/j.compmedimag.2019.05.001.

[47] Maithra Raghu et al. “Transfusion: Understanding transfer learning with applications to medical imaging”. In: *Proceedings of Neural Information Processing Systems (NeuriPS’19)*. 2019, pp. 3347–3357. DOI: 10.5555/3454287.3454588.

[48] Thomas Mensink et al. “Factors of Influence for Transfer Learning across Diverse Appearance Domains and Task Types”. In: *arXiv* (2021). DOI: arXiv:2103.13318.

[49] Jason Brownlee. *Better deep learning: train faster, reduce overfitting, and make better predictions*. Machine Learning Mastery, 2018.

[50] Chiyuan Zhang et al. “Understanding deep learning (still) requires rethinking generalization”. In: *Communications of the ACM* 64.3 (2021), pp. 107–115. DOI: 10.1145/3446776.

[51] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. "L2 Regularization for Learning Kernels". In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI'09)*. 2009, pp. 109–116. DOI: 10.5555/1795114.1795128.

[52] Pierre Baldi and Peter J Sadowski. "Understanding dropout". In: *Advances in neural information processing systems* 26 (2013), pp. 2814–2822.

[53] Patrice Simard et al. "Tangent Prop - A formalism for specifying selected invariances in an adaptive network". In: *Proceedings of Neural Information Processing Systems (NeuriPS'91)*. Vol. 91. 1991, pp. 895–903.

[54] Lutz Prechelt. "Early stopping-but when?" In: *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69. DOI: 10.1007/978-3-642-35289-8_5.

[55] Xiaojin Zhu and Andrew B Goldberg. "Introduction to semi-supervised learning". In: *Synthesis lectures on artificial intelligence and machine learning* 3.1 (2009), pp. 1–130. DOI: 10.2200/S00196ED1V01Y200906AIM006.

[56] Longlong Jing and Yingli Tian. "Self-supervised visual feature learning with deep neural networks: A survey". In: *IEEE transactions on pattern analysis and machine intelligence* 43 (2020), pp. 4037–4058. DOI: 10.1109/TPAMI.2020.2992393.

[57] Nitin Namdeo Pise and Parag Kulkarni. "A survey of semi-supervised learning methods". In: *Proceedings of the International conference on computational intelligence and security (ICCIS'08)*. Vol. 2. 2008, pp. 30–34.

[58] Dong-Hyun Lee et al. "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks". In: *Proceedings of the Workshop on challenges in representation learning, (ICML'13)*. Vol. 3. 2. 2013, p. 896. DOI: 10.1.1.664.3543.

[59] Qizhe Xie et al. "Self-training with noisy student improves imagenet classification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*. 2020, pp. 10687–10698. DOI: 10.1109/CVPR42600.2020.01070.

[60] Takeru Miyato et al. "Virtual adversarial training: a regularization method for supervised and semi-supervised learning". In: *IEEE transactions on pattern analysis and machine intelligence* 41.8 (2018), pp. 1979–1993. DOI: 10.1109/TPAMI.2018.2858821.

[61] Antti Tarvainen and Harri Valpola. "Mean Teachers Are Better Role Models: Weight-Averaged Consistency Targets Improve Semi-Supervised Deep Learning Results". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS17)*. 2017, pp. 1195–1204.

[62] Laine Samuli and Aila Timo. "Temporal ensembling for semi-supervised learning". In: *Proceedings of the International Conference on Learning Representations (ICLR'17)*. Vol. 4. 5. 2017, p. 6.

[63] Kihyuk Sohn et al. “FixMatch: Simplifying Semi-Supervised Learningwith Consistency and Confidence”. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS’20)*. Vol. 33. 2020, pp. 596–608.

[64] David Berthelot et al. “Mixmatch: A holistic approach to semi-supervised learning”. In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS’19)*. 2019, pp. 5050–5060. DOI: 10.5555/3454287.3454741.

[65] Longlong Jing and Yingli Tian. “Self-supervised visual feature learning with deep neural networks: A survey”. In: *IEEE transactions on pattern analysis and machine intelligence* 43 (2020), pp. 4037–4058. DOI: 10.1109/TPAMI.2020.2992393.

[66] Mehdi Noroozi and Paolo Favaro. “Unsupervised learning of visual representations by solving jigsaw puzzles”. In: *Proceedings of the European conference on computer vision (ECCV’16)*. Springer. 2016, pp. 69–84. DOI: 10.1007/978-3-319-46466-4_5.

[67] Ting Chen et al. “A simple framework for contrastive learning of visual representations”. In: *Proceedings of the International conference on machine learning (ICML’20)*. 2020, pp. 1597–1607.

[68] Lovedeep Gondara. “Medical Image Denoising Using Convolutional Denoising Autoencoders”. In: *Proceedings of the IEEE 16th International Conference on Data Mining Workshops (ICDMW’16)*. 2016, pp. 241–246. DOI: 10.1109/ICDMW.2016.0041.

[69] Aravindh Mahendran, James Thewlis, and Andrea Vedaldi. “Cross pixel optical-flow similarity for self-supervised learning”. In: *Proceedings of the Asian Conference on Computer Vision (ACCV’18)*. Springer. 2018, pp. 99–116. DOI: 10.1007/978-3-030-20873-8_7.

[70] Nikos Komodakis and Spyros Gidaris. “Unsupervised representation learning by predicting image rotations”. In: *Proceedings of the International Conference on Learning Representations (ICLR’18)*. 2018.

[71] Vassili Kovalev, Alexander Kalinovsky, and Sergey Kovalev. “Deep learning with theano, torch, caffe, tensorflow, and deeplearning4j: Which one is the best in speed and accuracy?” In: *Proceedings of the XIII International Conference on Pattern Recognition and Information Processing (PRIP’16)*. 2016, pp. 99–103.

[72] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

[73] Google. *Google Cloud: AuotML*. https://cloud.google.com/automl. 2022.

[74] Haifeng Jin, Qingquan Song, and Xia Hu. “Auto-Keras: An Efficient Neural Architecture Search System”. In: *Proceedings of the 25th International Conference on Knowledge Discovery & Data Mining (ACM SIGKDD’19)*. 2019, pp. 1946–1956. DOI: 10.1145/3292500.3330648.

[75] Aaron Klein et al. "Model-based Asynchronous Hyperparameter and Neural Architecture Search". In: *arXiv* (2020). DOI: arXiv:2003.10865.

[76] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. "Neural architecture search: A survey". In: *The Journal of Machine Learning Research* 20.1 (2019), pp. 1997–2017. DOI: 10.5555/3322706.3361996.

[77] Barret Zoph et al. "Learning Transferable Architectures for Scalable Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*. 2018, pp. 1–14. DOI: 10.1109/CVPR.2018.00907.

[78] Esteban Real et al. "Regularized Evolution for Image Classifier Architecture Search". In: *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI'19)*. Vol. 33. 2019, pp. 4780–4789. DOI: 10.1609/aaai.v33i01.33014780.

[79] Alex Krizhevsky and Geoffrey Hinton. *Learning multiple layers of features from tiny images*. Tech. rep. 0. University of Toronto, 2009.

[80] Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. "Sequential Model-Based Optimization for General Algorithm Configuration". In: *Proceedings of the 5th Learning and Intelligent Optimization (LION'11)*. Ed. by Carlos A. Coello Coello. 2011, pp. 507–523. DOI: 10.1007/978-3-642-25566-3_40.

[81] Chris Thornton et al. "Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms". In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD'13)*. 2013, pp. 847–855. ISBN: 9781450321747. DOI: 10.1145/2487575.2487629.

[82] Matthias Feurer et al. "Efficient and Robust Automated Machine Learning". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS'15)*. Vol. 28. 2015, pp. 2962–2970. DOI: 10.5555/2969442.2969547.

[83] Ryuichiro Hataya et al. "Faster autoaugment: Learning augmentation strategies using backpropagation". In: *Proceedings of the European Conference on Computer Vision (ECCV'20)*. Springer. 2020, pp. 1–16. DOI: 10.1007/978-3-030-58595-2_1.

[84] Milena Soriano Marcolino et al. "The impact of mHealth interventions: systematic review of systematic reviews". In: *JMIR mHealth and uHealth* 6.1 (2018), e23. DOI: 10.2196/mhealth.8873.

[85] Jiasi Chen and Xukan Ran. "Deep Learning With Edge Computing: A Review". In: *Proceedings of the IEEE* 107.8 (2019), pp. 1655–1674. DOI: 10.1109/JPROC.2019.2921977.

[86] Curtis T. Rueden et al. "ImageJ2: ImageJ for the next generation of scientific image data". In: *BMC Bioinformatics* 18 (2017), p. 529. DOI: 10.1186/s12859-017-1934-z.

[87] Fabrice de Chaumont et al. “Icy: an open bioimage informatics platform for extended reproducible research”. In: *Nature Methods* 9.7 (2012), pp. 690–696. DOI: 10.1038/nmeth.2075.

[88] Thouis R. Jones et al. “CellProfiler Analyst: data exploration and analysis software for complex image-based screens”. In: *BMC Bioinformatics* 9.1 (2008), p. 482. DOI: 10.1186/1471-2105-9-482.

[89] Aalfredo Canziani, Adam Paszke, and Eugenio Culurciello. “An Analysis of Deep Neural Network Models for Practical Applications”. In: *arXiv* (2016). DOI: arXiv:1605.07678.

[90] Bichen Wu et al. “Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR’19)*. 2019, pp. 10734–10742. DOI: 10.1109/CVPR.2019.01099.

[91] Han Cai et al. “Once-for-all: Train one network and specialize it for efficient deployment”. In: *Proceedings of the International Conference on Learning Representations (ICLR’20)*. 2020.

[92] Mark Sandler et al. “MobileNetV2: Inverted Residuals and Linear Bottlenecks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR’18)*. 2018, pp. 4510–4520. DOI: 10.1109/CVPR.2018.00474.

[93] Simon Wiedemann et al. “DeepCABAC: A Universal Compression Algorithm for Deep Neural Networks”. In: *IEEE Journal of Selected Topics in Signal Processing* 14.4 (2020), pp. 700–714. DOI: 10.1109/JSTSP.2020.2969554.

[94] Benoit Jacob et al. “Quantization and training of neural networks for efficient integer-arithmetic-only inference”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR’18)*. 2018, pp. 2704–2713. DOI: arxiv:1908.09791.

[95] Fei Li et al. “Development and clinical deployment of a smartphone-based visual field deep learning system for glaucoma detection”. In: *npj Digital Medicine* 3 (2020), p. 120. DOI: 10.1038/s41746-020-00329-9.

[96] Shorav Suriyal, Christopher Druzgalski, and Kumar Gautam. “Mobile assisted diabetic retinopathy detection using deep neural network”. In: *Proceedings of the Global Medical Engineering Physics Exchanges/Pan American Health Care Exchanges (GMEPE/PAHCE’18)*. 2018, pp. 1–4. DOI: 10.1109/GMEPE-PAHCE.2018.8400760.

[97] Saket S. Chaturvedi, Kajol Gupta, and Prakash S. Prasad. “Skin Lesion Analyser: An Efficient Seven-Way Multi-class Skin Cancer Classification Using MobileNet”. In: *Proceedings of the AMLTA 2020: Advanced Machine Learning Technologies and Applications (AMLTA’20)*. Vol. 1141. 2020, pp. 165–176. DOI: 10.1007/978-981-15-3383-9_15.

[98] Anliang Wang, Xiaolong Yan, and Zhijun Wei. "ImagePy: an open-source, Python-based and platform-independent software package for bioimage analysis". In: *Bioinformatics* 34.18 (Apr. 2018), pp. 3238–3240. DOI: 10.1093/bioinformatics/bty313.

[99] Google Brain team. *ImageJ/TensorFlow integration library plugin*. 2018. URL: https://imagej.net/TensorFlow.

[100] Google. *TensorFlow: An open-source software library for Machine Intelligence*. 2018. URL: https://www.tensorflow.org.

[101] Anne Carpenter. *CellProfiler 3.0 release: faster, better, and 3D*. 2018. URL: https://blog.cellprofiler.org/2017/10/16/cellprofiler-3-0-release-faster-better-and-3d/.

[102] Ignacio Arganda-Carreras et al. *Trainable Weka Segmentation plugin*. 2018. URL: https://imagej.net/Trainable_Weka_Segmentation.

[103] Wei Ouyang. *Rapid Learning - Icy plugin*. 2013. URL: http://icy.bioimageanalysis.org/plugin/Rapid_Learning.

[104] Michael R. Berthold et al. "KNIME: The Konstanz Information Miner". In: (2008). Ed. by Christine Preisach et al., pp. 319–326. DOI: 10.1007/978-3-540-78246-9_38.

[105] Estibaliz Gómez-de-Mariscal et al. "DeepImageJ: A user-friendly environment to run deep learning models in ImageJ". In: *Nature Methods* 18.10 (2021), pp. 1192–1195. DOI: 10.1038/s41592-021-01262-9.

[112] Alexander B. Jung et al. *imgaug*. 2020. URL: %5Curl%7Bhttps://github.com/aleju/imgaug%7D.

[113] Marcus D Bloice, Peter M Roth, and Andreas Holzinger. "Biomedical image augmentation using Augmentor". In: *Bioinformatics* 35.21 (Apr. 2019), pp. 4522–4524. DOI: 10.1093/bioinformatics/btz259.

[114] Alexander Buslaev et al. "Albumentations: Fast and Flexible Image Augmentations". In: *Information* 11.2 (2020). DOI: 10.3390/info11020125.

[115] Ekin D Cubuk et al. "Autoaugment: Learning augmentation strategies from data". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*. 2019, pp. 113–123. DOI: 10.1109/CVPR.2019.00020.

[116] Ruobing Huang, J. Alison Noble, and Ana I. L. Namburete. "Omni-Supervised Learning: Scaling Up to Large Unlabelled Medical Datasets". In: *Proceedings of the Medical Image Computing and Computer Assisted Intervention (MICCAI'18)*. 2018, pp. 572–580. DOI: 10.1007/978-3-030-00928-1_65.

[117] Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. "Model compression: making big, slow models practical". In: *Proceedings of the 12th International Conference on Knowledge Discovery and Data Mining (KDD'06)*. 2006, pp. 535–541. DOI: 10.1145/1150402.1150464.

[118] Jeremy Howard and Sylvain Gugger. *Deep Learning for Coders with Fastai and PyTorch: AI Applications Without a PhD*. O'Reilly Media, Inc., 2020.

[119] Leslie Smith. “Cyclical Learning Rates for Training Neural Networks”. In: *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV’17)*. 2017, pp. 464–472. DOI: 10.1109/WACV.2017.58.

[120] Kaggle. *APTOS 2019 Blindness Detection*. https://www.kaggle.com/c/aptos2019-blindness-detection. 2019.

[121] Daniel S. Kermany et al. “Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning”. In: *Cell* 172.5 (2018), 1122–1131.e9. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2018.02.010. URL: http://www.sciencedirect.com/science/article/pii/S0092867418301545.

[122] Marina Arredondo-Santoyo et al. “Automatic characterisation of dye decolourisation in fungal strains using expert, traditional, and deep features”. In: *Soft Computing* 23 (2019), pp. 12799–12812. DOI: 10.1007/s00500-019-03832-8.

[123] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. “The HAM10000 Dataset: A Large Collection of Multi-Source Dermatoscopic Images of Common Pigmented Skin Lesions”. In: *Scientific Data* 5 (Mar. 2018). DOI: 10.1038/sdata.2018.161.

[124] Noel C. F. Codella et al. “Skin Lesion Analysis toward Melanoma Detection: A Challenge at the International Symposium on Biomedical Imaging (ISBI) 2016”. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR’17)*. 2017.

[125] Konstantin Pogorelov et al. “KVASIR: A Multi-Class Image Dataset for Computer Aided Gastrointestinal Disease Detection”. In: *Proceedings of the 8th ACM on Multimedia Systems Conference (MMSys’17)*. 2017, pp. 164–169. DOI: 10.1145/3083187.3083212. URL: http://doi.acm.org/10.1145/3083187.3083212.

[126] Kaggle. *Open Sprayer images*. $https://www.kaggle.com/gavinarmstrong/open-sprayer-images$. 2019.

[127] Thomas Mosgaard Giselsson et al. “A Public Image Database for Benchmark of Plant Seedling Classification Algorithms”. In: *arXiv* arXiv:1711.05458 (2017).

[128] Jayant Kumar, Peng Ye, and David Doermann. “Structural similarity for document image classification and retrieval”. In: *Pattern Recognition Letters* 43 (2014), pp. 119–126. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2013.10.030.

[129] Afra Zomorodian. “Topological data analysis”. In: *Advances in applied and computational topology* 70 (2012), pp. 1–39. DOI: 10.1.1.440.3587.

[130] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. “Testing the manifold hypothesis”. In: *Journal of the American Mathematical Society* 29.4 (2016), pp. 983–1049. DOI: 10.1090/jams/852.

[131] Joe Davison. *DEvol - Deep Neural Network Evolution*. https://github.com/joeddav/devol. 2018.

[132] Piero Molino, Yaroslav Dudin, and Sai S. Miryala. "Ludwig: a type-based declarative deep learning toolbox". In: *arXiv* arXiv:1909.07930 (2019).

[133] Nikita Orlov et al. "WND-CHARM: Multi-purpose image classification using compound image transforms". In: *Pattern recognition letters* 29.11 (2008), pp. 1684–1693. DOI: 10.1016/j.patrec.2008.04.013.

[134] Lior Shamir et al. "IICBU 2008: A Proposed Benchmark Suite for Biological Image Analysis". In: *Medical & Biological Engineering & Computing* 46.9 (2008), pp. 943–947. DOI: 10.1007/s11517-008-0380-5.

[135] Mingxing Tan and Quoc V. Le. "MixConv: Mixed Depthwise Convolutional Kernels". In: *Proceedings of the 30th British Machine Vision Conference 2019 (BMVC'19)*. 2019, pp. 1–13.

[136] Mingxing Tan et al. "MnasNet: Platform-Aware Neural Architecture Search for Mobile". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*. 2019, pp. 2815–2823. DOI: 10.1109/CVPR.2019.00293.

[137] Hang Zhang et al. "ResNeSt: Split-Attention Networks". In: *arXiv* (2020). DOI: arXiv:2004.08955.

[138] Francois Chollet. *Deep Learning with Python*. Manning Publications, 2018.

[139] Yangqing Jia et al. "Caffe: Convolutional Architecture for Fast Feature Embedding". In: *Proceedings of the 22nd ACM international conference on Multimedia (ICMR'14)*. 2014. DOI: 10.1145/2647868.2654889.

[140] Eclipse Deeplearning4j Development Team. *Deeplearning4j: Open-source, Distributed Deep Learning for the JVM*. 2018. URL: https://deeplearning4j.org/.

[141] Tianqi Chen et al. "MxNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems". In: *In Proceedings of the Neural Information Processing Systems (NIPS 2015) - Workshop on Machine Learning Systems* (2015).

[142] Adam Paszke et al. "Automatic differentiation in PyTorch". In: *Proceedings of the Neural Information Processing Systems (NIPS'17)- Workshop*. 2017.

[143] Oliver Rittho et al. "Yale: Yet Another Learning Environment". In: *Proceedings of the Tagungsband der GI-Workshop-Woche Lernen– Lehren – Wissen Adaptivitat (LLWA'03)*. 2001, pp. 84–92.

[144] Rustam I. Aminov. "A brief history of the antibiotic era: lessons learned and challenges for the future". In: *Frontiers Microbiology* 8.1 (2010), p. 134. DOI: 10.3389/fmicb.2010.00134.

[145] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Proceedings of the Medical Image Computing and Computer-Assisted Intervention (MICCAI'15)*. 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

[146] Liang-Chieh Chen et al. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”. In: *Proceedings of the European Conference on Computer Vision (ECCV’18)*. Ed. by Vittorio Ferrari et al. 2018, pp. 833–851. DOI: 10.1007/978-3-030-01234-2_49.

[147] Kaiming He et al. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision (ICCV’17)*. 2017, pp. 2961–2969. DOI: 10.1109/ICCV.2017.322.

[148] Jingdong Wang et al. “Deep High-Resolution Representation Learning for Visual Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.10 (2021), pp. 3349–3364. DOI: 10.1109/TPAMI.2020.2983686.

[149] Xuebin Qin et al. “U2-Net: Going deeper with nested U-structure for salient object detection”. In: *Pattern Recognition* 106 (2020), p. 107404. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2020.107404.

[150] R. Y. Foos. “Vitreoretinal juncture — Simple epiretinal membranes”. In: *Albrecht von Graefes Archiv für klinische und experimentelle Ophthalmologie* 189 (1974), pp. 231–250.

[151] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR’16)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[152] Mingxing Tan and Quoc Le. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *Proceedings of the 36th International Conference on Machine Learning (ICML’19)*. Vol. 97. ICML. 2019, pp. 6105–6114.

[153] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *Proceedings of the International Conference on Learning Representations (ICLR’21)*. 2021.

[154] Hugo Touvron et al. “Training data-efficient image transformers& distillation through attention”. In: *Proceedings of the 38th International Conference on Machine Learning (ICML’21)*. Vol. 139. 2021, pp. 10347–10357.

[155] Jeremy Howard and Sylvain Gugger. “Fastai: A Layered API for Deep Learning”. In: *Information* 11 (Feb. 2020), p. 108. DOI: 10.3390/info11020108.

[156] Alireza Tavakkoli et al. “A novel deep learning conditional generative adversarial network for producing angiography images from retinal fundus photographs”. In: *Scientific Reports* 10.21580 (2020). DOI: 10.1038/s41598-020-78696-2.

[157] Jun-Yan Zhu et al. “Unpaired image-to-image translation using cycle-consistent ad-versarial networks”. In: *Proceedings of the International Conference on Computer Vision (ICCV’17)*. 2017, pp. 2242–2251. DOI: 10.1109/ICCV.2017.244.

[158] Liyuan Liu et al. “On the Variance of the Adaptive Learning Rate and Beyond”. In: *Proceedings of the International Conference on Learning Representations (ICLR’20)*. 2020.

[159] Michael Zhang et al. “Lookahead Optimizer: k steps forward, 1 step back”. In: *Proceedings of the Advances in Neural Information Processing Systems*. Vol. 32. 2019. DOI: 10.5555/3454287.3455148.

[160] Christian Szegedy et al. “Rethinking the inception architecture for computer vision”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR’16)*. 2016, pp. 2818–2826. DOI: 10.1109/CVPR.2016.308.

[161] Hongyi Zhang et al. “mixup: Beyond Empirical Risk Minimization”. In: *International Conference on Learning Representations (ICLR’18)*. 2018.

[162] Ilya Loshchilov and Frank Hutter. “SGDR: Stochastic Gradient Descent with Warm Restarts”. In: *Proceedings of the International Conference on Learning Representations (ICLR’17)*. 2017.

[163] Samiksha Pachade et al. “Retinal Fundus Multi-Disease Image Dataset (RFMiD): A Dataset for Multi-Disease Detection Research”. In: *Data* 6.2 (2021), p. 14. DOI: 10.3390/data6020014.

[164] Cha Zhang and Yunqian Ma, eds. *Ensemble Machine Learning: Methods and Applications*. Springer, 2012.

[165] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *Proceedings of the International Conference on Learning Representations (ICLR’15)*. 2015.

Publications

[106] Ángela Casado et al. “CLoDSA: a tool for augmentation in classification, localization, detection, semantic segmentation and instance segmentation tasks”. In: *BMC Bioinformatics* 20 (2019). DOI: 10.1186/s12859-019-2931-1.

[107] Adrián Inés et al. “Biomedical image classification made easier thanks to transfer and semi-supervised learning”. In: *Computer Methods and Programs in Biomedicine* 198 (2021), p. 105782. DOI: 10.1016/j.cmpb.2020.105782.

[108] Adrián Inés et al. “Semi-Supervised Learning for Image Classification using Compact Networks in the BioMedical Context”. In: *arXiv* (2022). DOI: arXiv:2205.09678.

[109] Adrián Inés et al. “DeepClas4Bio: Connecting bioimaging tools with deep learning frameworks for image classification”. In: *Computers in Biology and Medicine* 108 (2019), pp. 49–56. DOI: 10.1016/j.compbiomed.2019.03.026.

[110] Ángela Casado-García et al. “MotilityJ: An open-source tool for the classification and segmentation of bacteria on motility images”. In: *Computers in Biology and Medicine* 136 (2021), p. 104673. DOI: 10.1016/j.compbiomed.2021.104673.

[111] Ángela Casado-García et al. “Prediction of Epiretinal Membrane from Retinal Fundus Images Using Deep Learning”. In: *Advances in Artificial Intelligence* (2021), pp. 3–13. DOI: 10.1007/978-3-030-85713-4_1.