Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Active Deep Learning for Medical Imaging Segmentation

Degree's Thesis
Audiovisual Systems Engineering

Author: Marc Górriz Blanch
Advisors: Xavier Giró-i-Nieto, Axel Carlier and Emmanuel Faure

Universitat Politècnica de Catalunya (UPC)
2016 - 2017

Abstract

This thesis proposes a novel active learning framework capable to train effectively a convolutional neural network for semantic segmentation of medical imaging, with a limited amount of training labeled data. Our approach tries to apply in segmentation existing active learning techniques, which is becoming an important topic today because of the many problems caused by the lack of large amounts of data.

We explore different strategies to study the image information and introduce a previously used cost-effective active learning method based on the selection of high confidence predictions to assign automatically pseudo-labels with the aim of reducing the manual annotations.

First, we made a simple application for handwritten digit classification to get started to the methodology and then we test the system with a medical image database for the treatment of melanoma skin cancer. Finally, we compared the traditional training methods with our active learning proposals, specifying the conditions and parameters required for it to be optimal.

Resum

Aquesta tesi proposa un nou marc d'aprenentatge actiu capaç d'entrenar de forma efectiva una xarxa neuronal convolucional per la segmentació semàntica d'imatges mèdiques, a través d'una quantitat limitada d'instàncies d'entrenament. El nostre enfocament intenta introduir tècniques existents d'aprenentatge actiu en el camp de la segmentació, un tema poc tractat en l'actualitat a causa dels nombrosos problemes que poden ocasionar la falta de dades d'entrenament.

Explorem diferents estratègies per l'estudi de l'informació de la imatge, que ens permeten dissenyar un mètode rentable de selecció de les instàncies més optimes per l'entrenament del nostre sistema. Aquestes tècniques ens permeten introduir en segmentació el mètode actiu de cost efectiu, molt utilitzat en tasques de classificació, que es basa a utilitzar de manera iterativa prediccions d'alta confiança com pseudo etiquetes amb l'objectiu de reduir la quantitat d'anotacions manuals.

En primer lloc, desenvolupem una simple aplicació per a la classificació de dígits manuscrits per iniciar-nos amb la metodologia i després testegem el sistema amb una base de dades d'imatges mèdiques per al tractament del càncer de pell de melanoma. Finalment, comparem el model clàssic d'entrenament amb les diferents modalitats actives proposades, especificant les condicions i els paràmetres pertinents perquè aquestes siguin òptimes.

Resumen

Esta tesis propone un nuevo marco de aprendizaje activo capaz de entrenar de manera efectiva una red neuronal convolucional para la segmentación semántica de imágenes médicas, a través de una cantidad limitada de instancias de entrenamiento. Nuestro enfoque trata de introducir técnicas existentes de aprendizaje activo en el campo de la segmentación, tema poco tratado en la actualidad debido a los numerosos problemas que puede causar la falta de datos en el entrenamiento de sistemas con gran cantidad de parametros.

Exploramos estrategias para el estudio de la información de la imagen, que nos permiten diseñar un método rentable de selección de instancias óptimas para el entrenamiento de nuestro sistema. Estas técnicas nos permiten introducir en segmentación el método activo de coste efectivo, muy usado en clasificación, que trata de usar de manera iterativa predicciones de alta confianza como pseudo etiquetas, haciendo disminuir la cantidad de anotaciones requeridas de forma manual.

En primer lugar, desarrollamos una simple aplicación para la clasificación de dígitos manuscritos para iniciarnos con la metodología y luego testeamos el sistema con una base de datos de imágenes médicas para el tratamiento del cáncer de piel de melanoma. Finalmente, comparamos el modelo clásico de entrenamiento con las distintas versiones de entrenamiento activo propuestas, especificando las condiciones y los paramentaras necesarios para que estas sean óptimo.

Acknowledgements

First of all I would like to thank my three advisors, Xavier Giró-i-Nieto from Barcelona and Axel Carlier and Emmanuel Faure from Toulouse, for your patience and your proficiency attitude. I am aware that without their help I would not have been able to face this thesis "out of my home". Also special thanks also to Albert Gil for all the technical support provided along the development of this thesis.

I would also like to thank the VORTEX group and the excellent people there, Sonia Mejbri, Bastien Durix, Damien Mariyanayagam, Arthur Renaudeau, Clément, Vincent, Thibaut... very thanks to welcome me since the first day and to make my life so easy, friends.

Finally I would like to thank to my family and my friends for your support and your patience to make possible all my purposes, thanks for all.

Revision history and approval record

Revision	Date	Purpose
0	05/06/2017	Document creation
1	20/06/2017	Document revision
2	28/06/2017	Document revision
3	30/06/2017	Document approbation

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Marc Górriz Blanch	gbmarc@hotmail.es
Xavier Giró i Nieto	xavier.giro@upc.edu
Axel Carlier	axel.carlier@enseeiht.fr
Emmanuel Faure	emmanuel.faure@irit.fr

Written by:		Reviewed and approved by:		Reviewed and approved by:	
Date	05/06/2017	**Date**	30/06/2017	**Date**	30/06/2017
Name	Marc Górriz Blanch	**Name**	Xavier Giró i Nieto	**Name**	Axel Carlier
Position	Project Author	**Position**	Project Supervisor	**Position**	Project Supervisor

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Statement of purpose

In general, one of the major problems in medical diagnosis is the subjectivity of the specialist's decisions. More concretely, in the fields of medical imaging interpretation, the experience of the specialist can greatly determine the outcome of the final diagnosis. Manual methods of visualization can sometimes be very tedious, time-consuming and subject to errors on part of the interpreter. This has led the growing of intelligent image-based diagnostics as a support, being one of the most current research topics nowadays.

The emergence of deep learning paradigm working through neural networks followed by the recent advances in computational power have enabled the development of new intelligent diagnostics based on computer vision. These diagnostics are capable to analyze images, performing accurate segmentations, in order to detect the lesion areas and to make final decisions about the patient's health as the best of clinical eyes.

Nevertheless, only very deep convolutional neural networks with large amounts of trainable parameters are able to approach this kind of semantic segmentations and therefore, huge amounts of useful and labeled data are required to make the system converge while avoiding over-fitting. This may be a heavy handicap in the medical imaging field, where the human and logistic costs could make unfeasible to get large labeled datasets.

Active Learning (AL) is an established way to reduce this labeling workload in order to select in an iterative way, the most informative examples from a subset of unlabeled instances. This choice is based on a ranking of scores that can be computed from several methodologies from a model outcome. The chosen candidates are labeled and subsequently added to the training set. It has been previously shown that the training done using this active learning methodology is more efficient and can train a deep network faster and with fewer training samples than traditional semi-supervised learning methods. However although this would be adequate in most cases and being highly used in complex classification models, re-applying existing techniques could not be sufficient in segmentation where the information of small details can be essential in the final decision.

For all the above, the main contributions of this project are:

- The design and training of a framework for medical imaging semantic segmentation using deep neuronal networks and the Cost-Effective Active Learning (CEAL) methodology.
- The development of image information interpreters for medical imaging based on *Monte Carlo Dropout* for the analysis of the intrinsic network weight distribution.
- The development of an open sourced software package capable to perform all the experiments and with the possibility to further extensions.

1.2 Requirements and specifications

As already mentioned before, this thesis' main goal is to implement and adapt an updatable software oriented to perform semantic segmentation. The requirements of this project are:

- Design and train a convolutional neuronal network architecture for medical imaging semantic segmentation using Active Learning techniques to prevent overfitting with insufficient data resources.
- Possibility to adapt the trained model for incremental training using new labeled data in the future.

The framework chosen is *Keras* that uses *TensorFlow* as backend, providing high level abstraction of the *Google's* developed software-based framework widely used in deep learning applications. The package is compatible with *Python* $2.7+$, programming language that will be used to develop all the project. Due to the high computational requirements of the code and the impossibility to be executed in a conventional CPU computer, a GPU (Graphics Processing Unit) is required to train and evaluate the models. In addition to CUDA libraries developed by *NVIDIA* to compile and perform the parallel computations on the GPU. The software resources and power supply of the *NVIDIA GeForce GTX TITAN X GPUs* will be provided by *Image Processing Group* (*GPI*) from *UPC Barcelona*.

1.3 Methods and procedures

This project tries to offer a viable solution to adapt existing Active Learning methodologies in imaging semantic segmentation. The solution proposed is to follow the guidelines of Cost-Effective Active Learning used for classification, changing the image information analysis methods to carry out segmentation problems. The core of our methodology is the image uncertainty computation as an information measurement, calculating the variance of the network decisions using *Monte Carlo Dropout* on test time.

The project proposes an exhaustive data information study during the training process, in order to define a criteria to automatically select the best data instances for the model outcome to be labeled, optimizing the overall workload in the annotation process. Taking the uncertainty computation, the data is projected on a region diagram to study the intrinsic nature of the images and their impact in the training.

All the tests were done with the ISIC 2017 Challenge dataset [4] for Skin Lesion Analysis towards melanoma detection, splitting the training set into labeled and unlabeled amount of data to simulate the Active Learning problem with large amounts of unlabeled data at the beginning.

The first contribution of the project is to apply in segmentation the CEAL methodology that uses the complementary sample selection within the active learning procedure, recognizing the best predictions to be used as an automatic pseudo-labels with the aim to increase the amount of training data in each iteration.

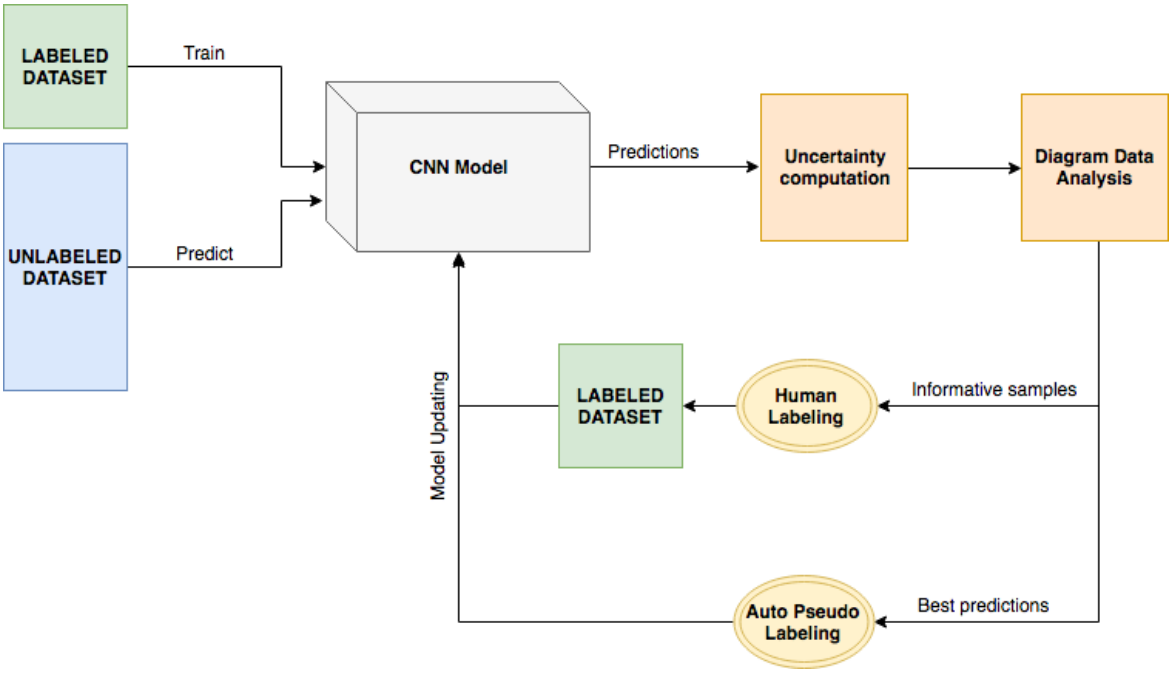


Figure 1.1: Overall methodology architecture proposed for this project

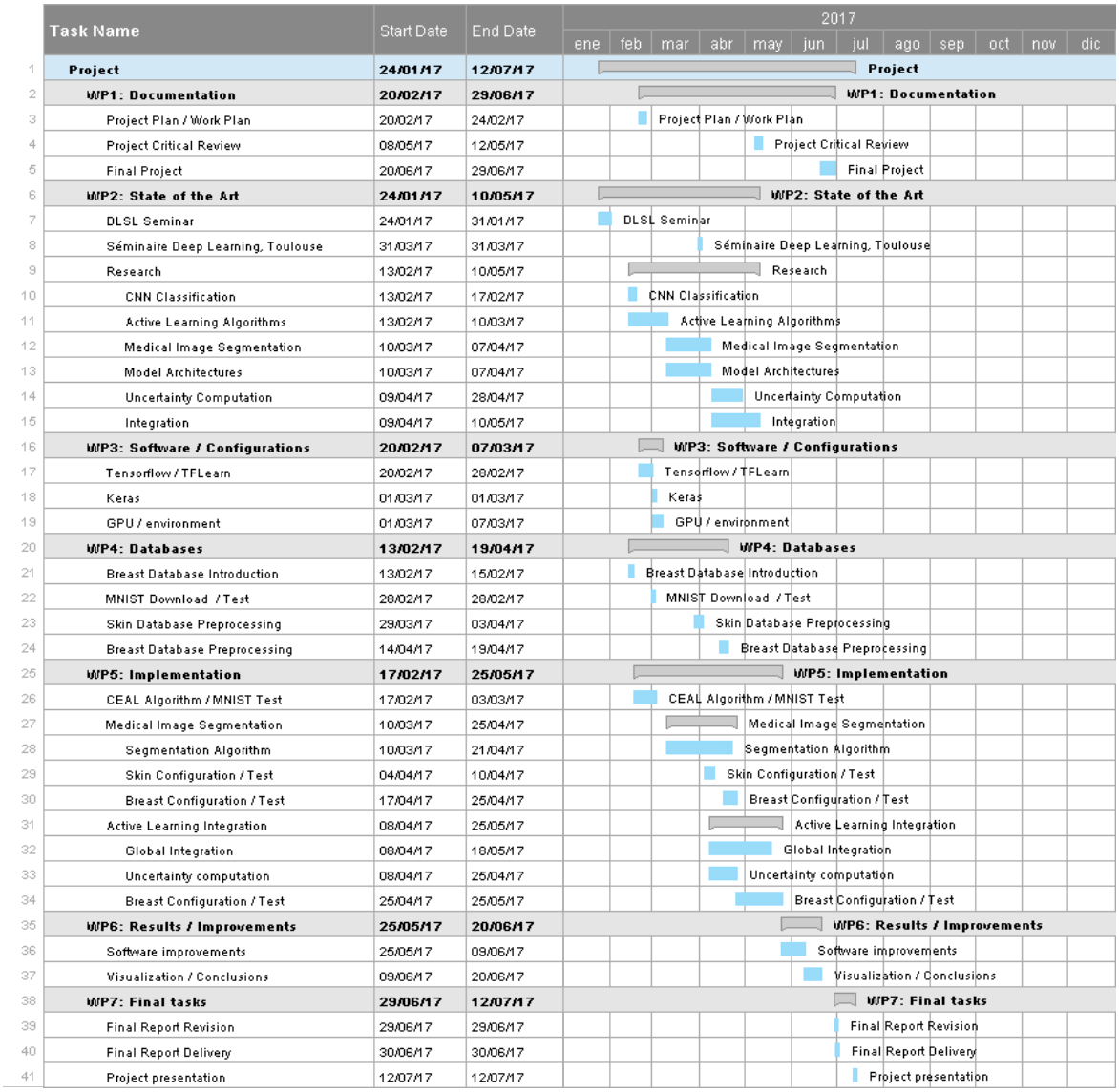
1.4 Work Plan

This project has followed the established work plan, with a few exceptions and modifications explained in the section [1.5.

1.4.1 Work Packages

- WP 1: Documentation
- WP 2: Research for the State of the Art
- WP 3: Software and Hardware configurations
- WP 4: Datasets
- WP 5: Implementation
- WP 6: Results and improvements
- WP 7: Final tasks

1.4.2 Gantt Diagram



	Task Name	Start Date	End Date
1	**Project**	**24/01/17**	**12/07/17**
2	**WP1: Documentation**	**20/02/17**	**29/06/17**
3	Project Plan / Work Plan	20/02/17	24/02/17
4	Project Critical Review	08/05/17	12/05/17
5	Final Project	20/06/17	29/06/17
6	**WP2: State of the Art**	**24/01/17**	**10/05/17**
7	DLSL Seminar	24/01/17	31/01/17
8	Séminaire Deep Learning, Toulouse	31/03/17	31/03/17
9	Research	13/02/17	10/05/17
10	CNN Classification	13/02/17	17/02/17
11	Active Learning Algorithms	13/02/17	10/03/17
12	Medical Image Segmentation	10/03/17	07/04/17
13	Model Architectures	10/03/17	07/04/17
14	Uncertainty Computation	09/04/17	28/04/17
15	Integration	09/04/17	10/05/17
16	**WP3: Software / Configurations**	**20/02/17**	**07/03/17**
17	Tensorflow / TFLearn	20/02/17	28/02/17
18	Keras	01/03/17	01/03/17
19	GPU / environment	01/03/17	07/03/17
20	**WP4: Databases**	**13/02/17**	**19/04/17**
21	Breast Database Introduction	13/02/17	15/02/17
22	MNIST Download / Test	28/02/17	28/02/17
23	Skin Database Preprocessing	29/03/17	03/04/17
24	Breast Database Preprocessing	14/04/17	19/04/17
25	**WP5: Implementation**	**17/02/17**	**25/05/17**
26	CEAL Algorithm / MNIST Test	17/02/17	03/03/17
27	Medical Image Segmentation	10/03/17	25/04/17
28	Segmentation Algorithm	10/03/17	21/04/17
29	Skin Configuration / Test	04/04/17	10/04/17
30	Breast Configuration / Test	17/04/17	25/04/17
31	Active Learning Integration	08/04/17	25/05/17
32	Global Integration	08/04/17	18/05/17
33	Uncertainty computation	08/04/17	25/04/17
34	Breast Configuration / Test	25/04/17	25/05/17
35	**WP6: Results / Improvements**	**25/05/17**	**20/06/17**
36	Software improvements	25/05/17	09/06/17
37	Visualization / Conclusions	09/06/17	20/06/17
38	**WP7: Final tasks**	**29/06/17**	**12/07/17**
39	Final Report Revision	29/06/17	29/06/17
40	Final Report Delivery	30/06/17	30/06/17
41	Project presentation	12/07/17	12/07/17

Figure 1.2: Gantt Diagram of the Degree Thesis

1.5 Incidents and Modification

The initial plan was to test different medical imaging datasets to get a extensible and adaptable software. But in order to understand and improve the Active Learning methodology it has been added a classification application out of the aim of the project and therefore to the initial plan. The practical classification application has modified the work plan, but in spite of this, it was possible the project in the agreed date, fulfilling all the expected objectives.

Chapter 2

State of the art

2.1 Convolutional Neural Networks

Deep convolutional neural networks (also known as CNN's or ConvNets) have recently become popular in computer vision, since they have dramatically advanced the state-of-the-art in tasks such as image classification [5], retrieval [6] or object detection [16] [12].

ConvNets are a type of feed-forward artificial neural network in which the connectivity pattern between its neurons, is inspired by the organization of the animal visual cortex. Individual cortical neurons respond to stimuli in a restricted region of space known as the receptive field. The receptive fields of different neurons partially overlap such that they tile the visual field. The response of an individual neuron to stimuli within its receptive field can be approximated mathematically by a convolution operation.

A CNN works similarly to Neural Networks: each neuron receive an input, a dot product (Hadamard product or elementwise multiplication) between each input and its associated weight is performed, followed with a non-linearity. The most common hierarchical distribution of ConvNets layers contains:

- Input layer, containing the raw pixel values from input images.
- Convolutional layers, the core block of ConvNets, computes a locally dot product (2D in the case of images) between the weights and a certain tiny region of the input volume.
- Non-linear layers, most of the times using a ReLU activation function which applies an elementwise activation by thresholding at zero.
- Pooling layers that apply a spatial downsampling along the output volume.
- Fully Connected layers that compute the class scores

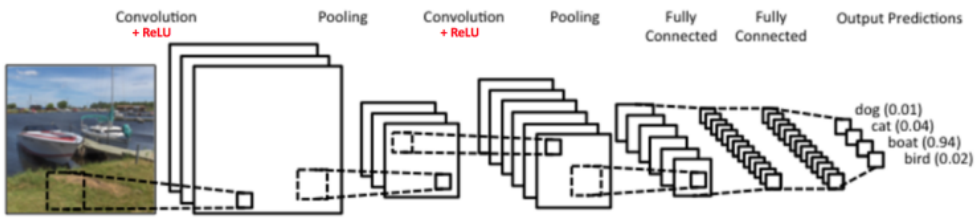


Figure 2.1: Convolutional Neural Network architecture

The learning process (also referred to network training) where weights are optimized is achieved through backpropagation [20], a technique to efficiently compute gradients for its weights with respect to the loss function.

2.1.1 Convolutional Neural Networks for Image Classification

Image classification is the task of taking an input image and outputting a class (a cat, dog, etc) or a probability of classes that best describes the image. For humans, this recognition task is one of the first skills we learn from the moment we are born and is one that comes naturally and effortlessly as adults. Without even thinking twice, we are able to quickly and seamlessly identify the environment we are in, as well as the objects that surround us. When we see an image or just when we look at the world around us, most of the time we are able to immediately characterize the scene and give each object a label, all without even consciously noticing.

Image classification has been one of the most important topics in the field of computer vision, trying to give to the machines the capability to recognize patterns, generalize from prior knowledge, and adapt to different image environments. Face recognition, vehicle detection, medical diagnosis and digit recognition are all excellent examples.

Convolutional neural networks have became a great revolution in this field, producing promising results. Examples of related work include:

- The work of Alex Krizhevsky, et al. [5] creating a "large, deep convolutional neural network" for image classification being the first model performing so well on a historically difficult ImageNet dataset. Utilizing techniques that are still used today, such as data augmentation and dropout.
- LeCun, Yann, et al. [17] reviews various methods applied to handwritten character recognition using CNN's. Related to this work, appears MNIST database (Modified National Institute of Standards and Technology database) a large dataset of handwritten digits widely used for training and testing in machine learning.
- S. Lawrence, et al. [16] explores the idea to use a ConvNet for face recognition in order to extract successively larger features in a hierarchical set of layers.
- B. Sahiner, Heang-Ping Chan, et al. [12] applies CNN's to medical imaging classifying regions of interest (ROI's) on mammograms as either mass or normal tissue.

2.1.2 Convolutional Neural Networks for Image Segmentation

Although ConvNets are widely used in classification, in many visual tasks, especially in biomedical image processing, the desired output should include localization, requiring an assignation of a class label to each pixel. This is the main idea of a semantic segmentation using ConvNets.

Recent semantic segmentation algorithms [14], convert an existing CNN architecture constructed for classification to a fully convolutional network (FCN). They obtain a coarse label map from the network by classifying every local region in image, and perform a simple deconvolution, which is implemented as bilinear interpolation, for pixel-level labeling. In addition, novel proposals [21][22], introduce the idea of deconvolution network to generate a dense pixel-wise class probability map by consecutive operations of unpooling, deconvolution, and rectification.

2.2 Active Learning

Usually, all of supervised and unsupervised learning tasks, first gather a significant quantity of data that is randomly sampled from the underlying population distribution and then induce a classifier or model. This methodology is called passive learning . Often the most hardly task in these applications is the labeling process in the data collection, since in many cases it must be manual annotated by an expert, being usually a time-consuming and costly task.

The emergence of semi-supervised machine learning proposes active learning methodology as a new solution in which a learning algorithm is able to interactively query the human annotator (or some other information source) new labeled instances from a pool of unlabeled data. Candidates to be labeled are chosen through several methods based on informativeness and uncertainty of the data for the intrinsic model distribution at any given moment.

All active learning algorithms follow the next key steps in an iteratively way: (a) **initialization**, the starting point before start learning process, pre-training with a starting labeled pool (b) **selection new training samples**, methodologies to choose new samples to be labeled based on pre-trained model, (c) **re-training**, training the model again adding the new labels to overall training set, and come back to new labels selection step, (d) **finalization**, define a methodology to stop the process.

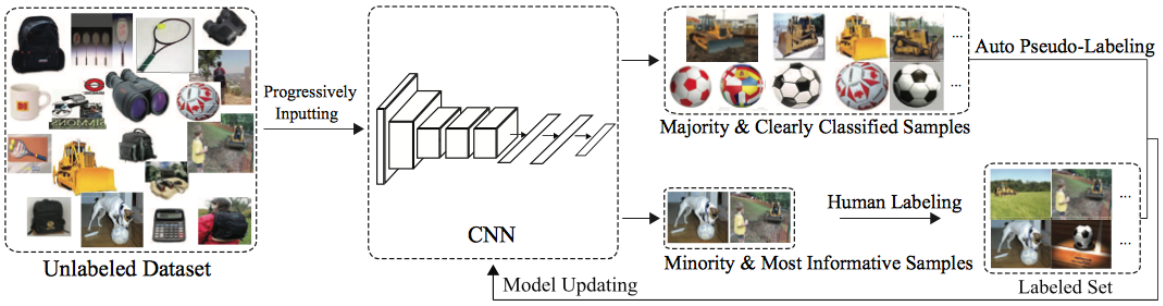


Figure 2.2: Cost-Effective Active Learning methodology [25], explained in Section 3.2

2.2.1 Active learning for Computer Vision

"Introduce the human in the loop" have become a popular saying within computer vision community. There are more and more applications exploring new strategies to interact to the user in order to get new labeled data.

Examples of real cases applied in image classification and segmentation includes:

- The work of Dhruv Batra, Adarsh Kowdle, et al. presenting *iCoseg* [8], an algorithm for Interactive Co-segmentation of a foreground object with Intelligent Scribble Guidance. The system interacts with the user deciding the most uncertain/informative image regions to be annotated, optimizing the learning performance.
- The recent work of Steve Branson, et al. is a great example of Active Learning application for image classification, presenting a Hybrid Human–Machine Vision System for

Fine-Grained Categorization [15]. A system composed of a human and a machine working together and combines the complementary strengths of computer vision algorithms and (non-expert) human users. The human users provide information of the object by clicks and answers to multiple choice questions. The machine intelligently selects the most informative question to pose to the user in order to identify the object class as quickly as possible. By leveraging computer vision and analyzing the user responses, the overall amount of human effort required, measured in seconds, is minimized.

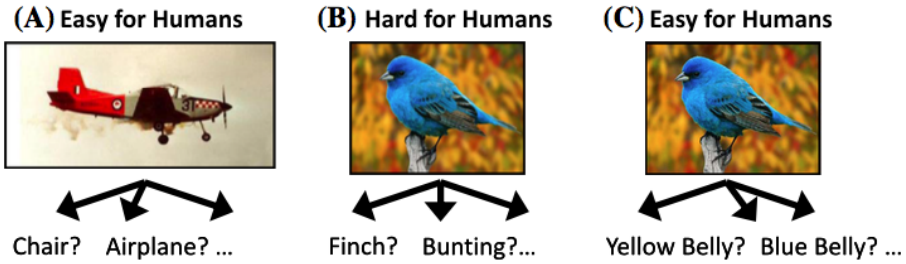


Figure 2.3: Examples of questions generated exploring the difficulty for the humans.

2.2.2 Active Deep Learning

One main concern of the deep learning community is to achieve deeper and deeper neuronal networks in order to increase their capacity of representation. Therefore, an incrementation of trainable parameters requires to scale up the size of the training database accordingly, falling once more upon the handicap to gather large amounts of labeled data. However, recent works [26] tend to show that deep learning may be handled with smaller dataset as long as the training samples are carefully selected, finding again active learning as a viable solution.

Currently, most of the existing algorithms on the state-of-the art are based on classification tasks [24][19]. The challenge of this project is to apply active learning methodologies in segmentation tasks. Being a real novel at the moment, in the way of perform pixel-wise criteria in the sample selection process and to escape from overfitting in the learning process.

Chapter 3

Active Learning methodology

3.1 Objective

The aim of this project is to train by active learning methodologies, a deep neural network to be able to perform semantic segmentation of medical images in order to isolate with detail the lesion areas within them. Interactive algorithms will be applied to achieve a competitive performance with a limited amount of labeled data. As a review, interactive training with reduced training datasets is few applied in segmentation due to the networks complexity.

Therefore, the following sections present existing active learning modalities on classification, studying the different possibilities to extend the framework to the desired segmentation task.

3.2 Cost-Effective Active Learning algorithm

In this section, we present an efficient existing algorithm for the proposed Cost-Effective Active Learning (CEAL) framework [25], which is enabled to train a ConvNet with sufficient unlabeled training data, overcoming the relative inconsistency between active learning and convolutional neural networks.

As a review, an active learning algorithm interacts with an external annotator during the learning process. Therefore, as shown at the Section 2.2, works by an iterative repetition of several key phases: initialization, selection new training data, re-train and finalization.

A - Initialization

Suppose we have a dataset of n samples denoted as $D = \{x_i\}_{i=1}^{n}$. We denote the currently annotated samples of D as D^L while the unlabeled ones as D^U. We suppose the most (or all) of them unlabeled, therefore before start there may origin two possible situations:

- **Only unlabeled data:** This case starts with the labeled pool D^L empty, forcing the oracle to make a pre-labeling, randomly selecting initial training samples from D^U and manually annotate them as the starting point. This could be hard to face in deep learning, due to the large amount of data needed to perform a useful training iteration.
- **Enough initial labeled data:** A certain amount of data to start is a relative parameter to verify, since it is usually correlated with the model complexity. Once defined, samples from each class are randomly chosen considering a complexity variety in order to prevent over-fitting in any specific category.

Initial labeled pool D^L will be used to initialize the convolutional neural network parameters W. We need to verify the training specific hyperparameters for this step such as the number of epochs, batch size or learning rate. (Appendices, Definition of hyperparameters 7.2)

B - Complementary sample selection

Fixing the network parameters W, the system must rank all the unlabeled data D^U, according to an active learning criteria based on the prediction confidence, to select two kinds of instances.

One kind is the minority samples with low prediction confidence that will be annotated by the oracle and posteriorly added to the overall labeled pool D^L. The other kind is the majority samples with high prediction confidence, called high confidence samples. For this set, the system will automatically assign pseudo-labels with no human labor cost, denoting them as D^H.

These two kinds of samples are complementary to each other for representing different confidence levels of the current model on the unlabeled dataset. Therefore, the correct management of them will have a essential impact on the overall outcome.

C - Re-Training

With all labeled data gathered $D^L \cup D^H$, we will update again the convolutional network weights W. Once again, we will need to verify the specific training parameters, not needing to be necessarily the same than the initial ones.

D - Finalization

All iterative algorithms need a stop criteria, in most of the cases, the process runs until reaching a predefined and verified maximum iteration T. However, the application on deep learning frameworks forces us to define new parameters to prevent known problematics for lack of data such as overfitting. Is common the verification of hyperparameters that control a maximum number of training epochs or an accuracy convergence before enter in an overfitting phase.

Algorithm 1 Cost-Effective Active learning algorithm

Input:
Unlabeled samples D^U, initially labeled samples D^L, low confidence selection size K_U, hight confidence selection size K_C, initial training epochs t_0, loop training epochs t_A.

Output:
CNN parameters W.

1: Initialize W with D^L, t_0 times.
2: **while** not reach *stop criteria* **do**
3: Rank all unlabeled data D^U, based on active learning criteria.
4: $D^L \leftarrow K_U$ low confidence samples, annotated by the oracle.
5: $D^H \leftarrow K_C$ hight confidence samples, annotated by the system itself.
6: Update W with $D^L \cup D^H$, t_A times.
7: **return** W

3.3 Active Learning methodology for classification tasks

Using the same terminology as defined on previous Section 3.2, suppose we have a dataset for classification of m categories and n samples denoted as $D = \{x_i\}_{i=1}^{n}$. The classifier will assign a category to each data instance. We denote the label of x_i as $y_i = j$, $j \in \{1, ..., m\}$, i.e., x_i belongs to the jth category.

The active learning criteria used to perform the complementary sample selection will be based on uncertainty sampling [18], selecting left in unlabeled pool D^U, the K_U most uncertain samples for oracle labeling and the K_C most certain ones for pseudo-labels assignation. The ranking will be based on the probability of each sample to belong on a specific category $p(y_i = j|x_i; W)$, considering the model state in each moment. The most common approaches includes:

1. Least confidence: Rank all unlabeled samples in an ascending order according to lc_i value, defined as:

$$lc_i = \max_j p(y_i = j|x_i; W), \qquad (1)$$

   If the probability of the most probable class for a sample is low then the classifier is uncertain about the sample.

2. Margin sampling: Rank all the unlabeled samples in an ascending order according to the msi value, defined as:

$$ms_i = p(y_i = j_1|x_i; W) - p(y_i = j_2|x_i; W), \qquad (2)$$

   where j_1 and j_2 represent the first and second most probable class labels predicted by the classifiers. The smaller of the margin means more uncertainty.

3. Entropy: Rank all the unlabeled samples in an descending order according to their en_i value, defined as:

$$en_i = -\sum_{j=1}^{m} p(y_i = j|x_i; W) \log p(y_i = j|x_i; W), \qquad (3)$$

   This method takes into account all class scores, hight entropy means hight uncertainty.

The amount of certain K_C and uncertain K_U samples selected from the ranking in each iteration, will be a key parameter to verify. During the training process, the prediction confidence will grow because of the uncertainty decrease, producing better and better predictions. Therefore, the amount of certain pseudo-annotations will follow this progression, being very low at the beginning and go increasing as the system goes achieving a better performance. On the other hand, the amount of uncertain samples will be verified depending on real world requirements such as the complexity of the samples and the human availability to label them. In general, the selection amount will be constant, although it could also be progressive up to a certain limit.

3.4 Active Learning methodology for segmentation tasks

As shown at the Section 2.1.2 a convolutional neuronal network for semantic segmentation must be able to make binary segmentation maps through pixel-wise predictions. Therefore, we will keep again the terminology used throughout this chapter, adding to the samples spatial dimensions in order to define their pixel-wise structure.

The active learning criteria used to perform the complementary sample selection is based on the intrinsic distribution of the unlabeled data, ranking the unlabeled pool D^U based on their influence on the model. Being D_x^U, D_y^U the unlabeled data and their labels respectively, we are interested in finding the posterior network distribution $p(W|D_x^U, D_y^U)$. In general, this posterior distribution is not tractable, therefore we need to approximate the distribution of these weights using variational inference [10]. This technique allows us to learn the distribution over the network's weights, $q(W)$, by minimizing the *Kullback-Leibler (KL)* divergence between this approximating distribution and the full posterior one:

$$KL(q(W)||p(W|D_x^U, D_y^U)), \qquad (4)$$

In [13] it is explored the possibility to use *Monte Carlo Dropout* to approximate the variational weights distribution $q(W)$ term. We suppose a ConvNet architecture composed of L layers indexing all of its weighs, taking into account their depth, by K x K matrices. For one single layer i each indexed index will be denoted as $k_{i,j}$, $j \in \{1, ..., K_i\}$, $i \in \{1, ..., L\}$. The dropout works by randomly deactivating network activations, therefore if dropout is applied, each index $k_{i,j}$ will be modeled through a *Bernulli* distribution:

$$k_{i,j} = m_{i,j} * b_{i,j}, \qquad (5)$$

$$b_{i,j} \sim \begin{cases} Bernulli(p_i) & i = j \\ 1 & i \neq j \end{cases}, \qquad (6)$$

being p_i the dropout probability and $m_{i,j}$ the weight index value without dropout.

On the other hand, in [9] it was shown that minimizing the cross-entropy loss objective function has the effect of minimizing the *Kullback-Leibler* divergence term. Therefore training the network with stochastic gradient descent will allow to introduce this approach on our methodology.

With all the above, the introduction of *Dropout* during the training will prevent overfitting, while will allow us to compute the pixel-wise sample uncertainty on test time. Being I_x a image pixel, we can estimate the uncertainty of its predicted label $\widetilde{I}_y$ computing the variance of T different predictions on the same pixel by the effect of *Dropout* through the network weights:

$$uncert(\widetilde{I}_y) = \frac{1}{T-1}\sqrt{\sum_{t=1}^{T}(\widetilde{I}_{y,t} - \widetilde{I}_y)}\ , \qquad (7)$$

$$\widetilde{I}_y = \frac{1}{T}\sum_{t=1}^{T}\widetilde{I}_{y,t}, \qquad (8)$$

In algorithmic terms, we apply the above proposal to compute the uncertainty of one data instance following the next pseudo-code:

Algorithm 2 *Dropout* uncertainty computation

Input:
Unlabeled data instance I, CNN parameters W with *Dropout* layers, *Dropout* steps T, *Dropout* probability p_d.

Output:
Pixe-wise uncertainty U_{PW}, overall uncertainty U.

1: Initialize P empty. ▷ 3D Structure to index all *Dropout* step predictions.
2: **while** not reach T **do**
3: $P \leftarrow \widetilde{I}_t$ step prediction using W with p_d in *Dropout* layers.
4: $U_{PW} = var(P)$
5: $U = sum(U_{PW})$
6: **return** U_{PW}, U

The precision of pixel-wise uncertainty maps will depend on the *Dropout* steps T and the *Dropout* probability p_d. Hight p_d means hight variation of network weights making difficult a consistent result with a finite number of step predictions. As was shown in [13] the ideal p_d value is 0.5 and a maximum precision will be obtained when $T \to \infty$.

In Chapter 5 is shown the application of this methodology in the practical case of medical imaging semantic segmentation, as an advance here it is shown an example computing the uncertainty map of a melanoma skin cancer image, using a convolutional neuronal network already trained. For this example, it is not important to know the nature of the image database or the network used, just to understand in a graphical way, all methodology described above:

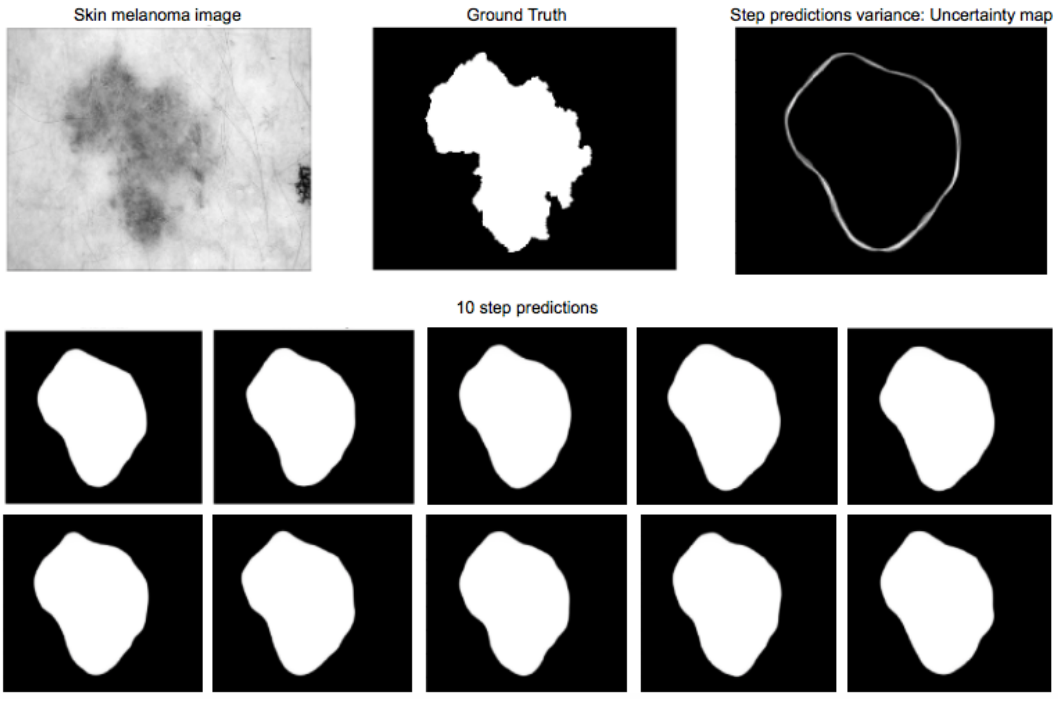


Figure 3.1: Pixel-wise uncertainty U_{PW} computation using $T = 10$ step predictions

Chapter 4

Practical Classification Application

The project starts with a pre-application of the active learning methodology on a classification task, designing and training a convolution neural network able to perform handwritten digit recognition. The objective is to be initiated into active learning framework trough a practical and simple toy application. The implementation and the subsequent results will help to define an order of magnitude of the parameters that will later be extended on the medical imaging semantic segmentation task.

4.1 MNIST database

The MNIST database was constructed out of the original NIST database; hence, modified NIST or MNIST. There are 60,000 training images (some of these training images can also be used for cross-validation purposes) and 10,000 test images, both drawn from the same distribution. All these black and white digits are size normalized, and centered in a fixed-size image where the center of gravity of the intensity lies at the center of the image with 28x28 pixels. [17] This is a relatively simple database containing 10 categories, that represent the numerical digits.

4.2 Convolutional Neural Network architecture

The network architecture is illustrated in Figure 5.1 and explained in Section 2.1 (Convolutional Neural Networks). The input layer of the network expects a 28x28 pixel gray image. The input image is passed through two convolutional blocks, including a 2D convolutional filters with a receptive field of 5x5 pixels, followed by a ReLU (nonlinearity operation) activation layer and a spatial max-pooling layer, performing a 2x2 subsampling. The network is concluded with a classifier block consisting of two Fully-Connected (FC) layers with 1024 and 10 neurons respectively.

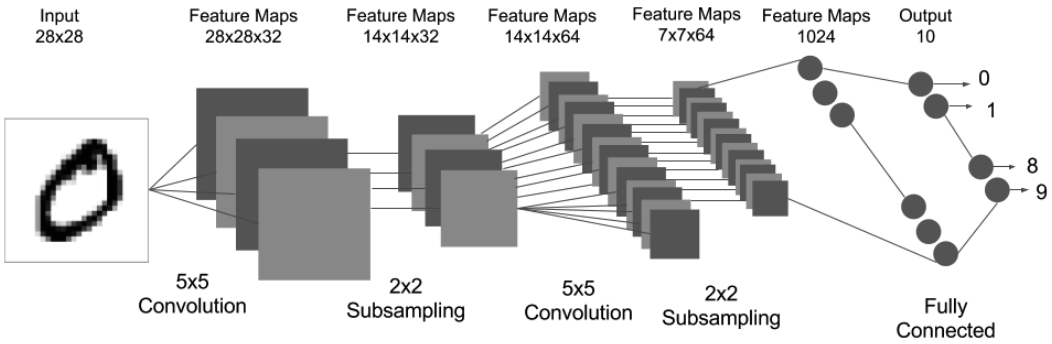


Figure 4.1: ConvNet architecture for handwritten digit recognition

The output FC layer is equipped with a Sigmoid activation in order to obtain a ranged score of each class. This probability, $p(y_i = j|x_i; W)$ will be used for the sample selection approach.

4.3 Experimental setup

The ConvNet architecture proposed to classify the MNIST handwritten digits, will be trained using the Cost-Effective Active Learning methodology explained in Section 3.2, with its specific modalities for classification tasks explained in Section 3.3. The aim of this chapter implementing a practical application, is to play with the parameters defined in each phase of the algorithm, seeing directly their effects on the system evaluation.

Since there are many parameters and infinite possible situations to test in each phase, we propose values for each parameter choosing a model that it has been tested during the implementation for be the most standard, with the aim to fix all the parameters while we are testing a specific one. The parameters terminology are explained in Section 3.2.

Step	Parameter		Value	Selection
Initialization	Initial D^L size		50	
			200	
			600	X
			2000	
			4000	
	Training epochs		1	X
			2	
			3	
Complementary Sample Selection	K_U size		0	
			20	
			50	
			200	X
			2000	
	K_C size	Constant	0	
			50	
			200	X
			600	
			2000	
	Method		Least Confidance	
			Margin Sampling	
			Entropy	X
Re - Training	Training epochs		1	X
			2	
			3	

Table 4.1: Classification experimental setup

The metric used to evaluate the classification experiments were the accuracy:

$$accuracy = \frac{number\ of\ correct\ predictions}{number\ of\ samples}, \qquad (9)$$

4.4 Results

This section describes the results of the experiments performed in each CEAL algorithm step. All the experiments follow the table above to initialize all the parameters.

4.4.1 Initialization experiments

The first experiment were related to the size of the initial labeled set. This parameter splits the MNIST training set, first selecting randomly the amount of labeled data to initialize D^L and then deleting the rest of the labels to initialize D^U. The evaluation has been based on MNIST test set but reduced being proportional to amount of training samples. The Figure 4.2 illustrates the accuracy values after initial train for different initial D^L sizes. Naturally, the more data we have the better result we will get but in the real world will depend on the nature of the problem and the real gathered data. However, we need to consider that few initial data could produce erroneous predictions for the pseudo annotations on the first interactions.

Another experiment was performed around the initial training epochs. Note in Figure 4.3 that one training epoch is enough to handle the standard model due to the simplicity of the data.

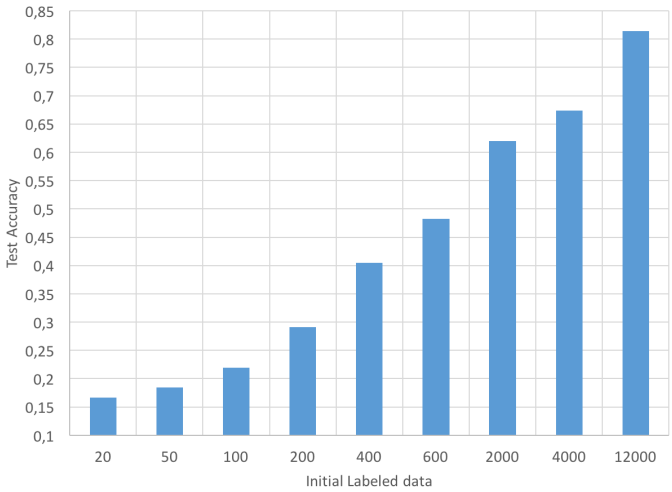


Figure 4.2: Initial evaluation for different D^L sizes

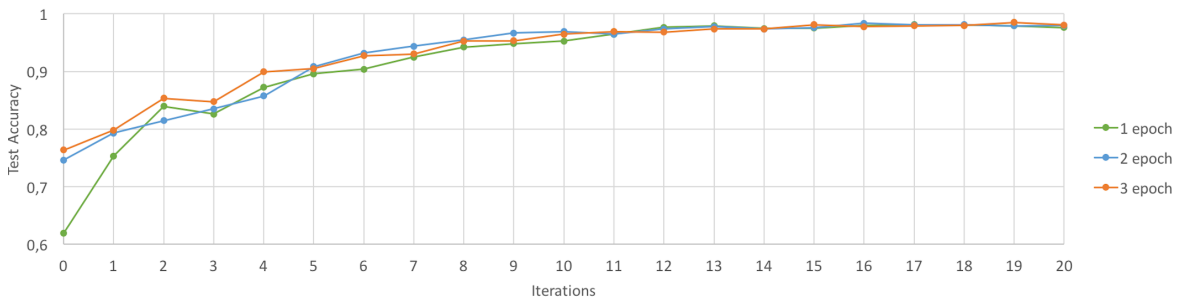


Figure 4.3: CEAL evaluation depending on the initial training epochs

4.4.2 Complementary sample selection experiments

The first experiments were related to the complementary sample selection. This is the most important step of the Cost-Effective Active Learning algorithm and its parameters will influence the overall system performance.

First was evaluated the sample selection method, Figure 4.4, choosing among the several solutions shown in Section 3.3. All the methods presents a similar performance, but it is chosen the entropy version in the standard model, following the CEAL state-of-the art [25]. In Figure 4.5 it is illustrated the entropy histogram in 2 interactions. Note the evolution of the uncertainty, being at the beginning the most of the instances located around a hight entropy value, and decreasing in the second figure, verifying therefore the correct performance of the learning process.

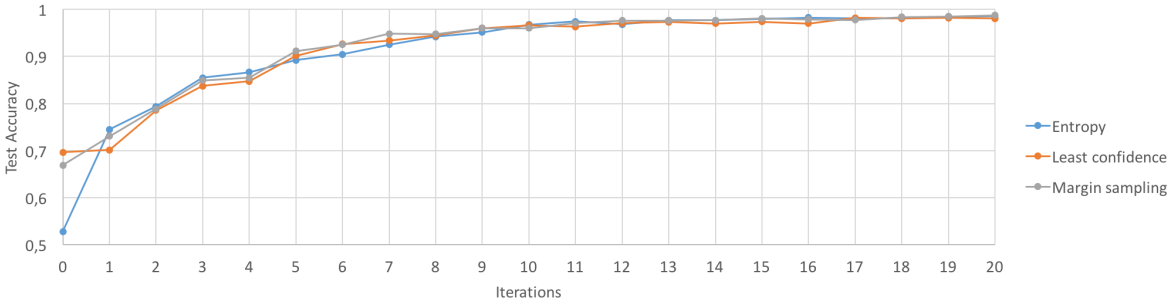


Figure 4.4: Complementary sample selection methods evaluation

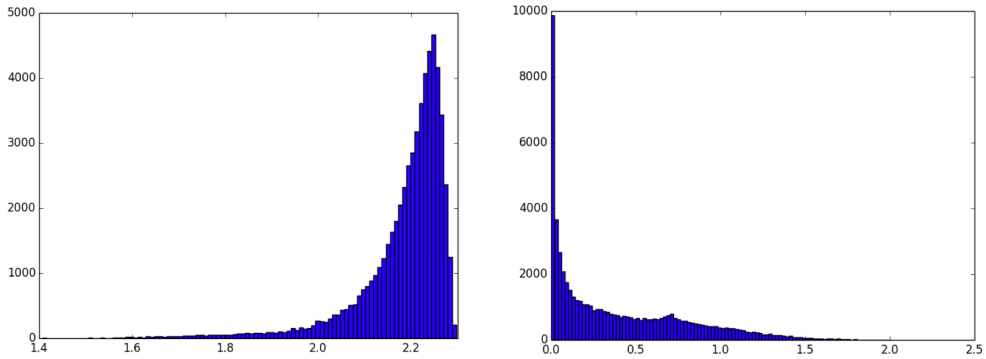


Figure 4.5: Uncertainty evolution during the interaction

The second experiment was related to the influence of the annotator during the interaction. Note the substantial improvement using the oracle annotations through the most informative samples. The pseudo-annotations are useful to complement them but they could be erroneous at the beginning. Next, it was evaluated the amount of manual and pseudo annotations in each iteration. In Figures 4.6 and 4.8 it is shown the system performance for different amounts of data. Note that the total of manual annotations is a decisive parameter while the pseudo-annotations have no relevance in this problem, but could be useful to complement certain problems when the data resources are very limited.

In order to see the effects of the amount of manual annotations in the iteration, in Figure 4.7 is illustrated the amount of active training iterations needed to spend the same labeled data than the 2000/iteration active model. Taking into account that an accuracy improvement is only achieved after a model update, the 20/interaction model needs around 60 training epochs to

achieve the same performance than the 2000/iteration model in only 5 epochs. However, it is interesting to observe that 1000 samples will be enough for all the models, therefore between 200 and 800 samples/interaction there are the models with an optimum balance between the amount of data and the needed interactive iterations.

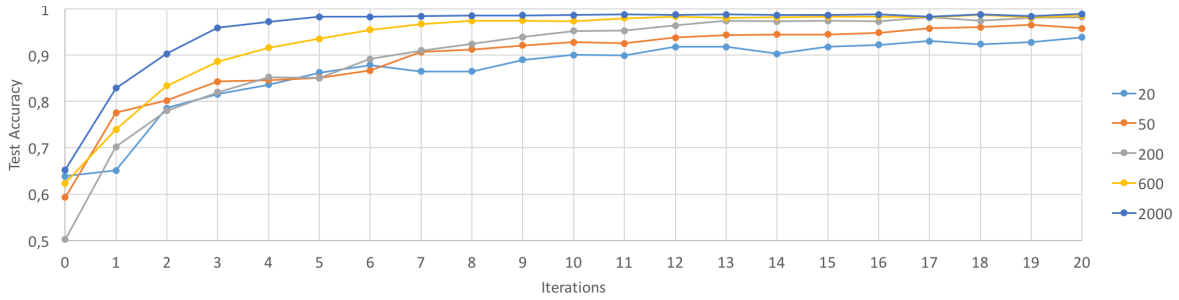


Figure 4.6: Amount of oracle annotations evaluation

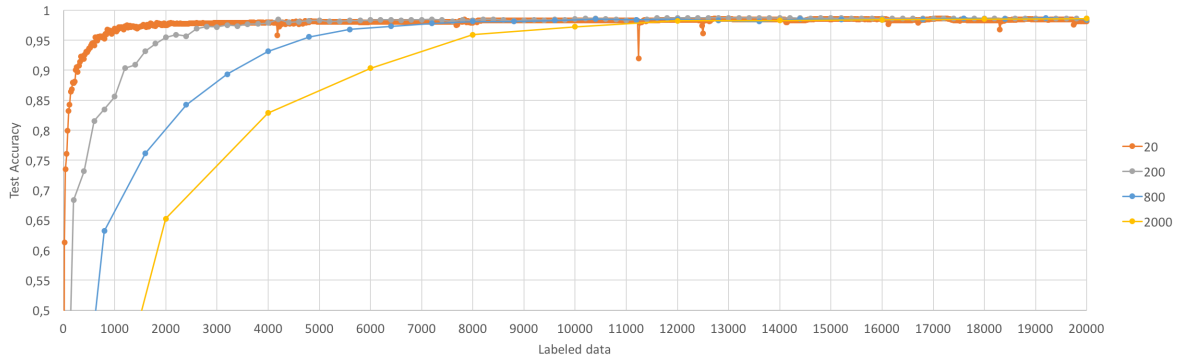


Figure 4.7: Effects of manual annotations in the interaction

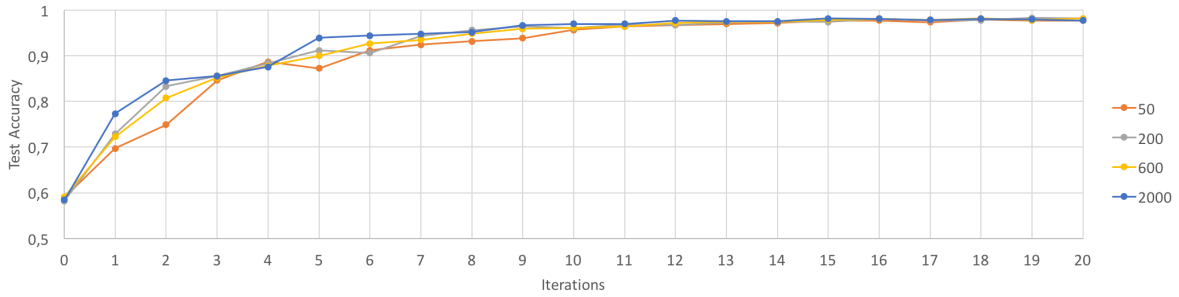


Figure 4.8: Amount of pseudo annotations evaluation

Finally it was been compared the classical method, using all the MNIST labeled training set, with the active learning interaction, using the parameters defined in the standard model.

Total manual annotations (+600 D^L size)	0	400	600	1200	2000	2600
% MNIST training set (40,000 samples)	1.5 %	2.5 %	3 %	5 %	6.5 %	8 %
% Classical training accuracy	52 %	80 %	85 %	90 %	95 %	99 %

Table 4.2: Classical training for classification comparative

Seeing the results we can conclude that it is possible to train a deep neuronal network for image classification with active learning methodologies, achieving a similar convergence with only the 8 % of the labeled samples.

Chapter 5

Medical Imagining Segmentation

The main contribution of this work is the development of an active learning methodology to be applied in an imaging segmentation system. By training a well-known Convolutional Neural Network architecture for semantic segmentation, applying the methods presented in the Section 5.2, this chapter discuses the practical application in medical imaging segmentation, defining the solutions that best fits the nature of the problem, achieving the best possible performance.

5.1 ISIC Dataset: Melanoma Skin Cancer

The ISIC 2016 Challenge dataset [4] for Skin Lesion Analysis towards melanoma detection was used for this work as a possible kind of medical data instances.

The dataset is publicly available and contains 2000 RGB dermoscopy images manually annotated by medical experts, by manual tracing the lesion boundaries in the form of a binary mask. The dataset was modified for this work, transforming the original images to gray scale and modifying their aspect ratio to adapt to the convNet input requirements.

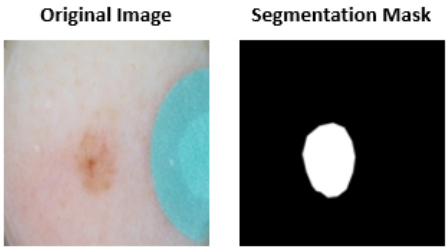


Figure 5.1: ISIC Archive Dataset example

5.2 U-Net architecture

The U-Net [23] is a convolutional neural network architecture specifically designed by O. Ronneberger et al. from the University of Freiburg to solve Biomedical Image Segmentation problems. It was successfully rated for winning the ISBI cell tracking challenge [1] 2015.

The network architecture is illustrated in Figure 5.2. As explained in Section 2.1.2, the network merges a convolutional network architecture with a deconvolutional architecture to obtain the semantic segmentation. The convolutional network is composed of a repetitive pattern of two 3 x 3 convolutions operations, followed by a ReLU layer and a downsampling process through a 2 x 2 maxpooling operation with stride 2.

On the other hand, the deconvolutional architecture includes a upsampling operation of the feature map obtained during the contracting path, followed by a 2×2 deconvolution that fractions the feature map channels into 2. A posteriori concatenation of the resulting feature map and the obtained during the contracting path is needed, followed by a 3×3 convolutions and a ReLU layer. The entire network is 23 convolutional layers deep, where the last layer is used to map each component feature vector related to the number of classes.

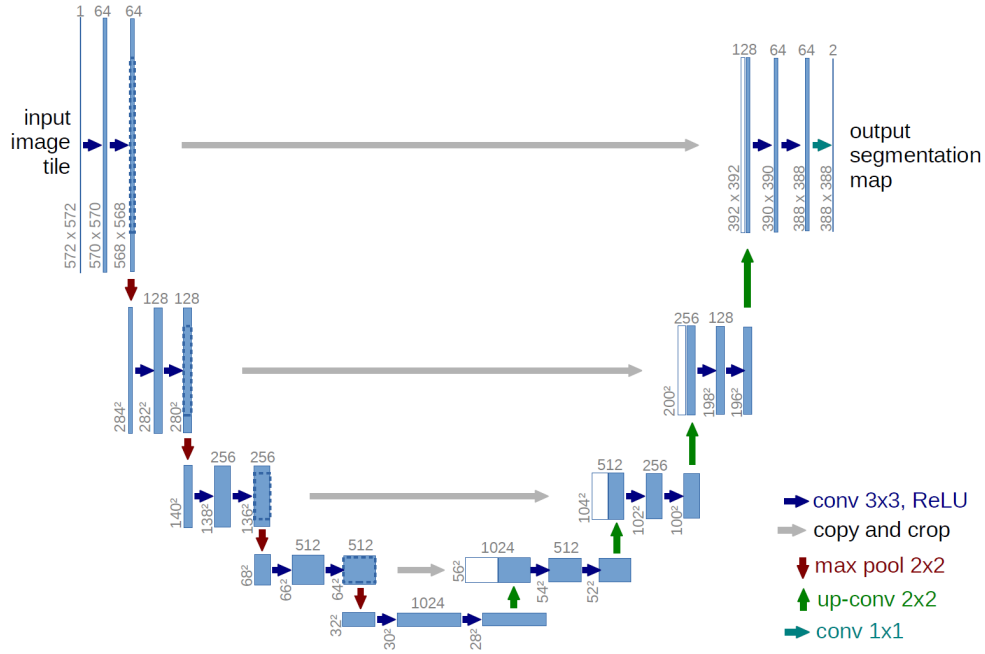


Figure 5.2: U-Net architecture (example)

5.3 Training parameters

The Section 3.4 related to the Active Learning methodology, shows that training the ConvNet learning weights with *Stochastic Gradient Descent (SGD)* in order to minimize the *cross-entropy* loss function, has the same effect than minimize the *Kullback-Leibler* divergence term, that tries to approximate the intractable posterior network distribution $p(W|D_x^U, D_y^U))$ with the full weights distribution $q(W)$, approximated in that case by *Monte Carlo Dropout*.

However, in this work it is used the *Dice Coefficient* loss function because of its pixel-wise discrimination for segmentation. The coefficient compares the pixel-wise agreement between the ground truth (Y) and its corresponding predicted segmentation ($\widetilde{Y}$).

$$dice\ coef = \frac{2 * |Y \cap \widetilde{Y}|}{|Y| + |\widetilde{Y}|}, \qquad (10)$$

$$dice\ loss\ function = -dice\ coef, \qquad (11)$$

The results prove that the *Dice Coefficient* loss function can be also used instead of *cross-entropy*, without breaking the correlation between the sample uncertainty and the system performance, considering thus, the weights learning difficulties for the image nature.

5.4 Initialization

Before starting the interactive learning process, the training sets are initialized based on the Cost-Effective Active Learning methodology. First, it is randomly selected the labeled amount D^L from the ISIC dataset and then the other labels are deleted initializing the unlabeled set D^U.

In addition, few previous preprocessing techniques are needed to adapt the data to the convolutional neuronal network architecture:

- **Image boundaries transformation**. The original images are preprocessed using *OpenCV* library, by cropping the images to the same aspect ratio and adapting their size to 192 x 240 following the U-Net input requirements.
- **Image channels reduction**. ConvNet input layer has an only depth map, requiring a gray scale transformation to reduce the original number of channels.
- **Mean subtraction**. In order to center the cloud of values from input data around zero, a mean subtraction is applied across the image features.
- **Image normalization**. By dividing each input image by its standard deviation, a normalization is obtained from its original 0 and 255 pixel values to 1 and 0 normalized values. This technique is commonly used in computer vision to avoid contrast issues.

5.5 Data augmentation

One powerful solution at the beginning of the CEAL methodology is the data augmentation. A widely used technique in matching learning to generate more training instances, performing basic transformations with the initial dataset. This methods help the initial training to achieve highest accuracy in less training epochs, in order to start the complementary sample selection with the best possible generalization ability, preventing further overfitting issues.

The transformations are performed by the *data generator* framework from *Keras*, generating random sets of new training data through the following transformations: rotations, horizontal and vertical shifts and horizontal flips.

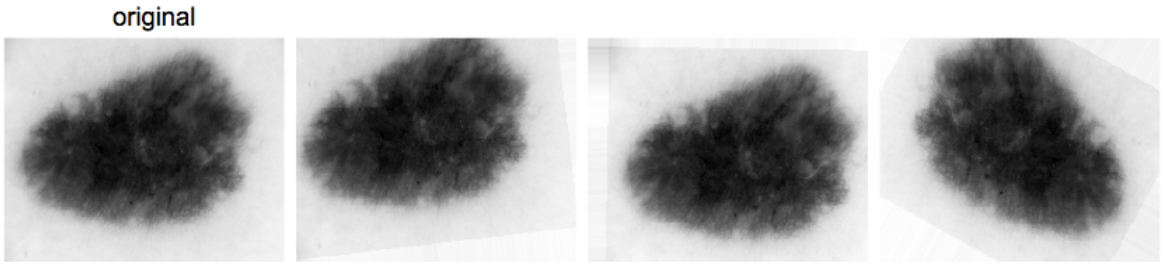


Figure 5.3: Data augmentation preview: random data generation by basic transformations.

5.6 Complementary Sample Selection

In order to select the most informative samples to be annotated, it is ranked all the unlabeled data by computing the overall uncertainty of each instance following the method shown in Section 3.4. The best ranking should achieve a perfect correlation between the uncertainty and the evaluation metric for the system, performing bad predictions the most uncertainty samples. As a review, the overall value it is computed by adding all pixel-wise values from the uncertainty map, therefore the samples containing more doubtful pixels, will achieve a highest final value.

This methodology is not always consistent with the nature of the images and their size. ISIC dataset is based on skin images with hight concentrations of melanin shaping structured cells. The size of the cells has a hight variance and it could determine the results. Figure 5.5 shows a real example where two samples with different cell size but with the same prior uncertainty, the bigger cell get hight uncertainty because of its large boundary extension.

To solve the problem it is performed a size normalization by the *euclidean distance transform* [11] of the uncertainty maps. The idea seems redundant but basically consists on make more thicker the thicker contours and more thinner the thinner ones. To do that, it is calculated the distance map of the sample prediction, by computing the euclidean distance of each pixel to the closest obstacle pixel (border pixels), being higher at the cell center and lower near the contours. Multiplying the distance map with the uncertainty pixels, it is obtained a new uncertainty map where the furthest pixels are penalized, getting size-normalized the size boundaries.

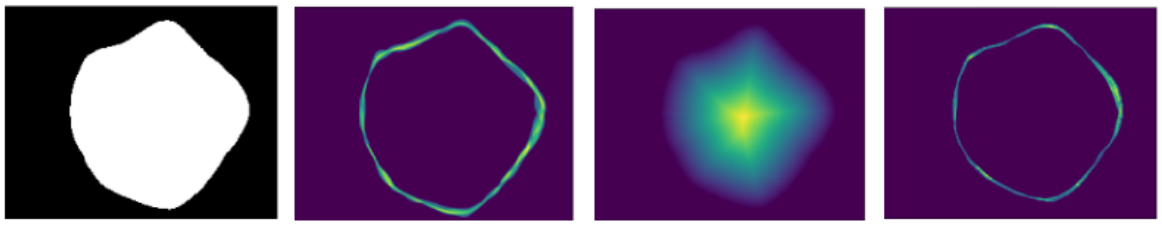


Figure 5.4: Size normalization summary: (a) Image prediction, (b) Uncertainty map, (c) Distance transform map, (d) Size-normalized uncertainty map: product between distance transform map and uncertainty map.

Once all the overall uncertain values are computed, we can use the ISIC training ground truth (deleted labels in the unlabeled set creation) to perform an experiment to visualize the correlation between the predictions goodness and their overall uncertain values. Remember that we used the word accuracy as the prediction goodness, although we are using the *Dice coefficient* as the evaluation metric.

In Figure 5.6 it is distinguished four regions according to the sample nature:

1. The samples in this regions have null accuracy while the uncertainty is low but variate, this samples are no-detected by the system, and although they have low uncertainty are more informative than the highest uncertain ones to be first manually annotated.
2. This region contains the hight uncertain samples, other possible candidates to be annotated by the oracle.

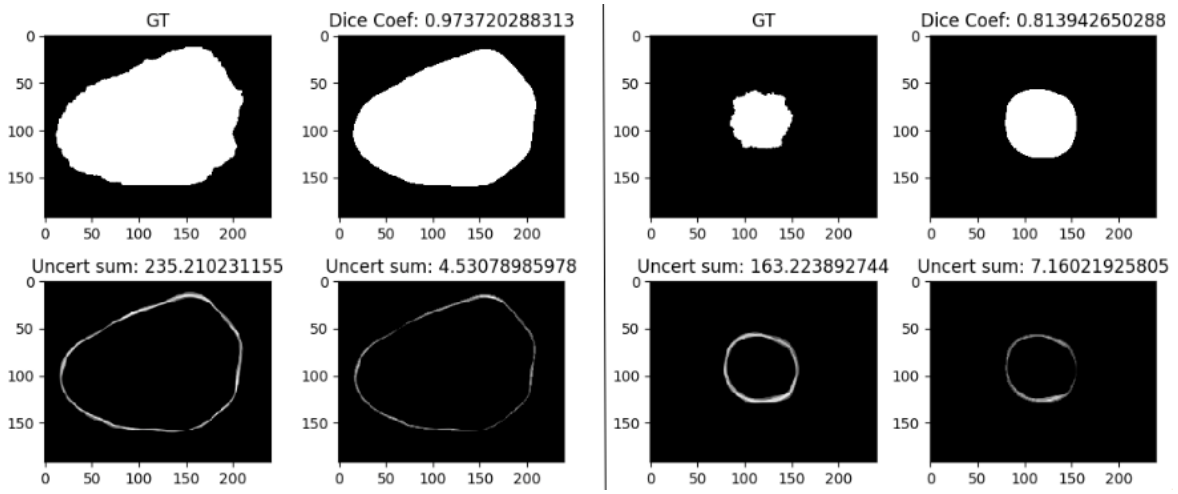


Figure 5.5: Size normalization example: This figure illustrates the size correlation problem with two different cells. For each one it is shown the ground truth (GT), its prediction with the evaluation metric (Dice coefficient), and on the bottom the uncertainty map with and without size normalization (left to right) with the respective overall uncertainty. Note the effect after applying the size normalization, being the second cell most uncertain in spite of its small size.

3. This region concentrates the most of the samples, there are located the highest accurate predictions with less uncertain values. These are perfect candidates to be selected as a pseudo-labels.

4. The central region contains a random amount of samples that will difficult the complementary sample selection.

In the real world, the system will only know the uncertain values and this region representation it wont be able, therefore the sample selection must be based only on the uncertainty axis, using the histogram of the uncertain values to get the overall distribution. Seeing the histogram in Figure 5.6, we can perform the region representation for one dimension to define the sample selection criteria. In the lowest side there will be the samples located on the first region, selecting K_1 samples around the highest bins to be manually annotated. Next, in the highest side there will be the samples located on the second region, selecting K_2 samples around the highest bins to be also manually annotated.

Then, in the central part, there will be the supposed samples to be pseudo annotated, but the instances in the fourth region will interfere to get good candidates from the thirst region. To solve that we will consider to select randomly K_4 samples from all unlabeled set, to be labeled by oracle, with the aim to reduce the concentration of bad predictions in the central area and to be able to select good K_C pseudo candidates in further iterations. This amount is expected to be incremental considering the progressively system improvement: $K_C = K_{C0} + i * K_{Ci}$, being K_{C0} the initial certain amount and K_{Ci} the amount for the active learning interaction i.

In summary, the complementary sample selection will gather the next labeled data: $K_U = K_1 + K_2 + K_4$ samples for oracle labeling and K_C samples for pseudo labeling. Following the Cost-Effective Active Learning methodology shown in Section 3.2, the K_U samples will be added to the labeled set D^L and the K_C will be added to an auxiliary set D^H used for the re-training but returned to the unlabeled set D^U again for the next complementary sample selection.

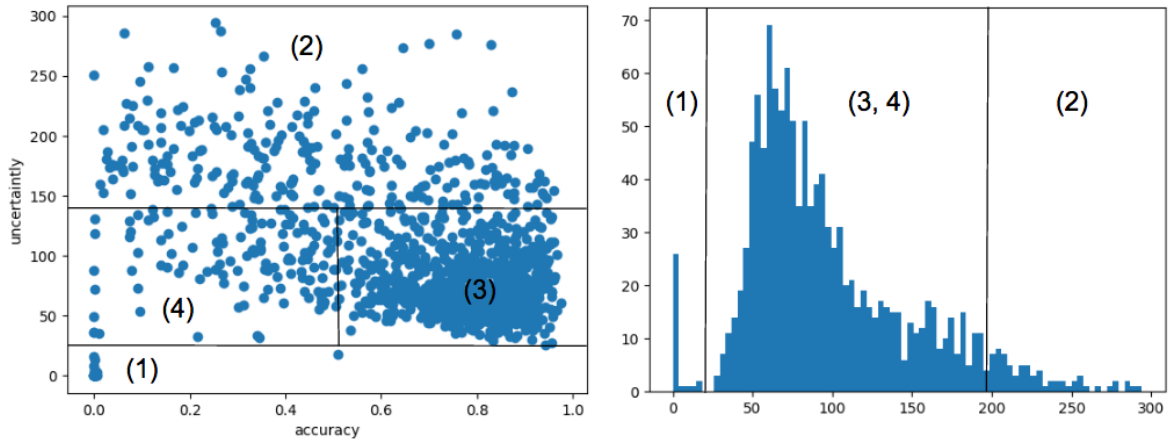


Figure 5.6: Regions diagram representation: comparative with the real world sampling selection criteria. Note the interference between regions 3 and 4 to choose the pseudo labeling candidates.

5.7 Experimental setup

Following the same methodology used in the classification experimentation, we defined a standard model of parameters to be fixed in all the experiments. The complexity of the problem could origin correlation between parameters being difficult to test all the possibilities. Despite all we proposed the following standard.

Step	Parameter		Value
Initialization	Inital D^L		600
	Training epochs		10
	Data augmentation		X
Complementary Sample Selection	Oracle annotations	K_1	10
		K_2	10
		K_4	10
		Starting epoch	0
	Pseudo annotations	Initial size	20
		Rate	20 $*$ iteration
		Starting epoch	5
	Dropout Step predictions		20
	Distance transform		X
Re-Train	Training epochs		2
Finalization	Iterations		20

Table 5.1: Segmentation experimental setup

The metric used in all the evaluations is the *Dice coefficient* shown in equation (10).

ISIC training dataset was split to gather a test set of 400 samples, therefore we only disposed of 1600 samples for the train that was split again to shape the initial labeled D^L and unlabeled D^U sets. Dice index was computed through the evaluation average of all the test data.

The following experiments were performed:

- Experiment 1. Initialization step. Evaluate the initialization parameters: the initial D^L size, the training epochs and the use of data augmentation to improve the initial point.
- Experiment 2. Complementary data selection. Effect of oracle labeling for no-detection K_1 sets.
- Experiment 3. Complementary data selection. Effect of oracle labeling for the most uncertain K_2 and random K_4 sets.
- Experiment 4. Complementary data selection. Effect of the pseudo labeling.
- Experiment 5. Comparative between the best achieved active learning performance and the classical training in terms of labeled data and the system performance.

5.8 Results

This section presents the results of the experiments on the medical imaging segmentation system described in this Chapter, based on the Cost-Effective Active Learning methodologies for segmentation presented in Section 3.2.

5.8.1 Initialization experiments

The first experiment was related to the initial training. This step is crucial for the system outcome, since it defines the starting point for the interactive loop. First we evaluated the number of training epochs in Figure 5.8, noting that unlike in classification tasks, they are more determinant due to the system complexity, needing more epochs but having a hight risk to converge in a bad performance and to disable the effect of further active learning iterations. Then we evaluated the effect of apply data augmentation in Figure 5.7, noting that it makes the system converge faster and with better performance, but it requires a high computational time for the augmentation procedures.

The evaluation of the D^L size it was not possible due the computational time. However, we had certain criteria about the magnitude order information of the classification task. For this project we chosen a initial labeled set of 600 samples but note that as same as classification this chose only depends on the real world application. Of course the more data we have the better performance we will get but we need to consider again the nature of the problem and the possible gathered amount of initial labeled data.

5.8.2 Effect of oracle labeling for no-detection K_1 set

In Section 5.6 was shown the importance of the K_1 set within the complementary sample selection methodology. As a review, no-detections are the most informative instances to be labeled by the oracle since they do not produce any stimulus to the ConvNet. Note in Figures 5.9 and 5.10 that we obtained great results training only with this kind of samples, but the regions

diagram does not converge in the expected way, having large amounts of data in the central and highest regions. Seeing in the Figure 5.7 a initial training convergence, it has been selected the CNN model corresponding to 9 epoch to save unless trainings in order to avoid overfitting. The following Figures show the system evaluation and the region diagrams for 10 iterations, using the parameter values defined in Table 5.1.

Initial Labeled Data	600 samples
30 annotations/iteration	300 samples
Total (10 iterations)	**900 samples**

Table 5.2: Experiment 1: Total of labeled data

5.8.3 Effect of oracle labeling for the most uncertain K_2 and random K_4 sets

The proposed solution to decrease the concentration of samples in the central and highest regions, in order to improve the pseudo sample selection is to select for the oracle labeling, the most uncertain K_2 and random K_4 sets together with the no-detections K_1.

In the results of regions graph for the first experiment in Figure 5.10, we observe a not optimal performance at the latest iterations, being assigned the most of the oracle candidates in the certain region. In this experiment we tried to introduce the sets progressively giving the K_1 more importance at the beginning and increasing the K_4 one at the end by reducing the no-detections. Note in Figure 5.9 the wanted convergence, increasing the overall performance. In summary, we followed the procedure defined in the following Table:

	Iteration 0 - 4	Iteration 5 - 9	Partial Total
Initial Labeled Data	600 samples		
K1 annotations/iteration	10 samples	5 samples	75 samples
K2 annotations/iteration	10 samples		100 samples
K4 annotations/iteration	10 samples	15 samples	125 samples
Total (10 iterations)			**900 samples**

Table 5.3: Experiment 2: Total of labeled data

5.8.4 Effect of pseudo annotations

This experiment were related to the effect of the pseudo-labels in the interaction. In the last sections we described the best possible environments to introduce this kind of annotations trying to concentrate the most of the samples at the certain region, increasing the system performance before this step. The results are satisfactory, but not as much as desired, noting in Figure 5.11 that there are still samples in the central region that interfere with the system improvement.

Nonetheless, it remains an open door for future works to continue researching new and more adapted solutions to achieve better starting points for the pseudo-labels, with the desire to see their potential power in this field.

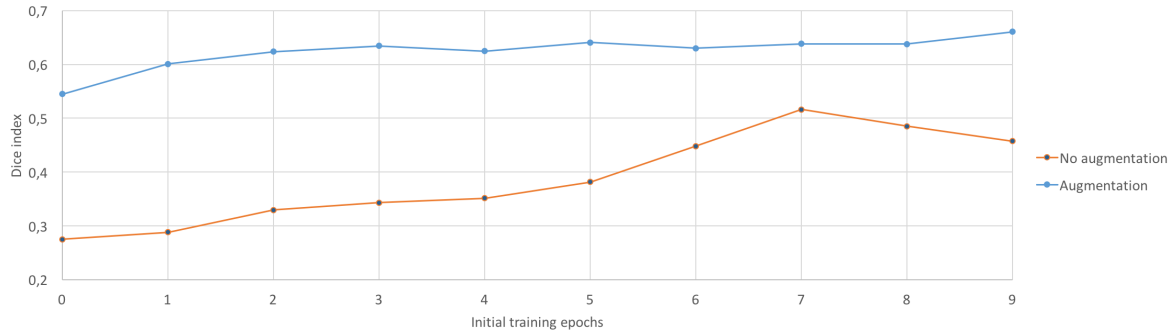


Figure 5.7: Initial training with data augmentation evaluation

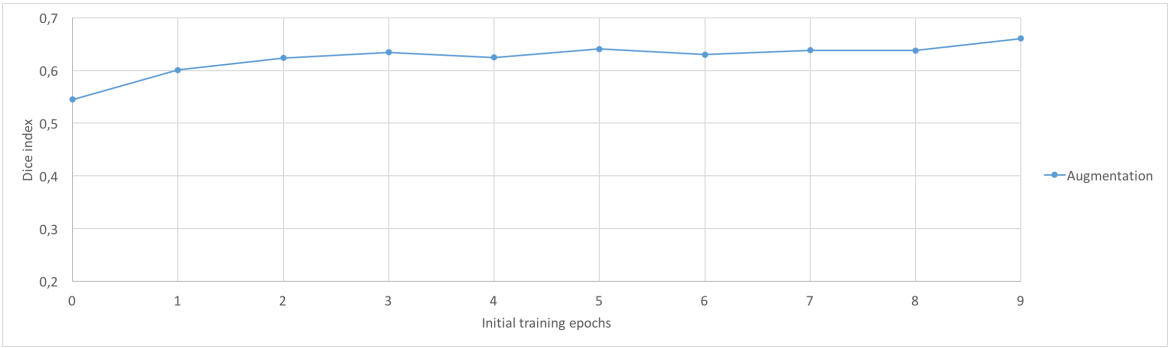


Figure 5.8: Initial training epochs evaluation.

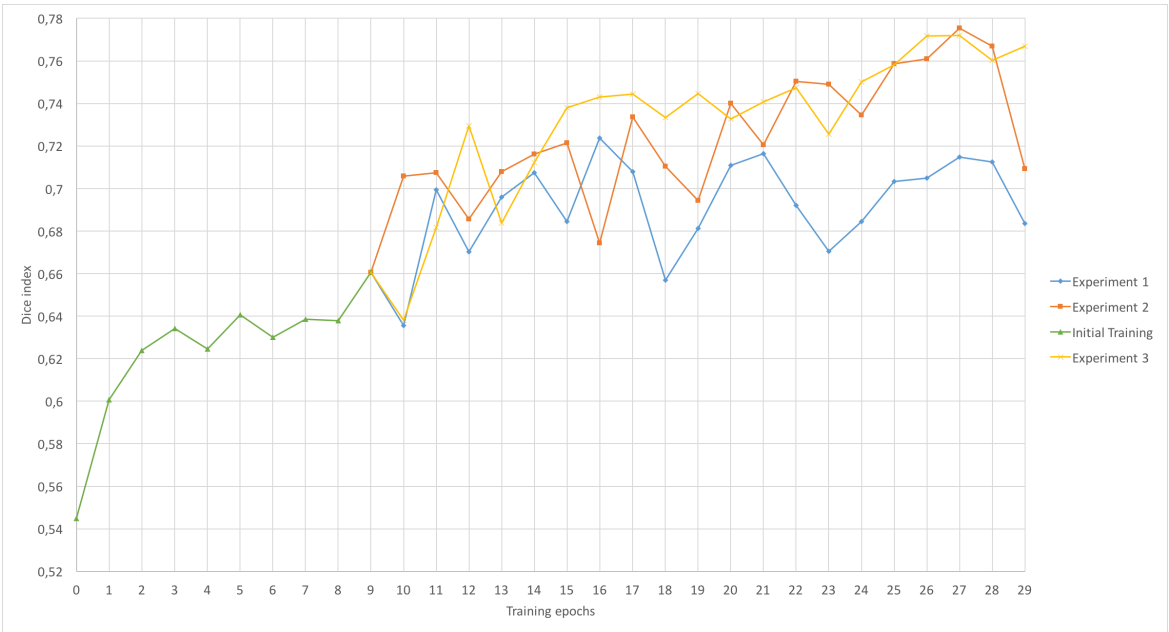


Figure 5.9: Experiments evaluation results. In the graph it is also included the initial train using data augmentation. Note that after the initial train, there was used 2 training epochs per iteration, being represented 10 active iterations in the graph. As a review: Experiment 1: Effect of oracle labeling for no-detection K_1 set; Experiment 2: Effect of oracle labeling for the most uncertain K_2 and random K_4 sets; Experiment 3: Effect of pseudo annotations. Note in Experiment 2, the effect of sets $K2$ and $K4$ after epoch 5 and in the Experiment 3, the effect of pseudo-labels improving the system but without achieve the expected performance.

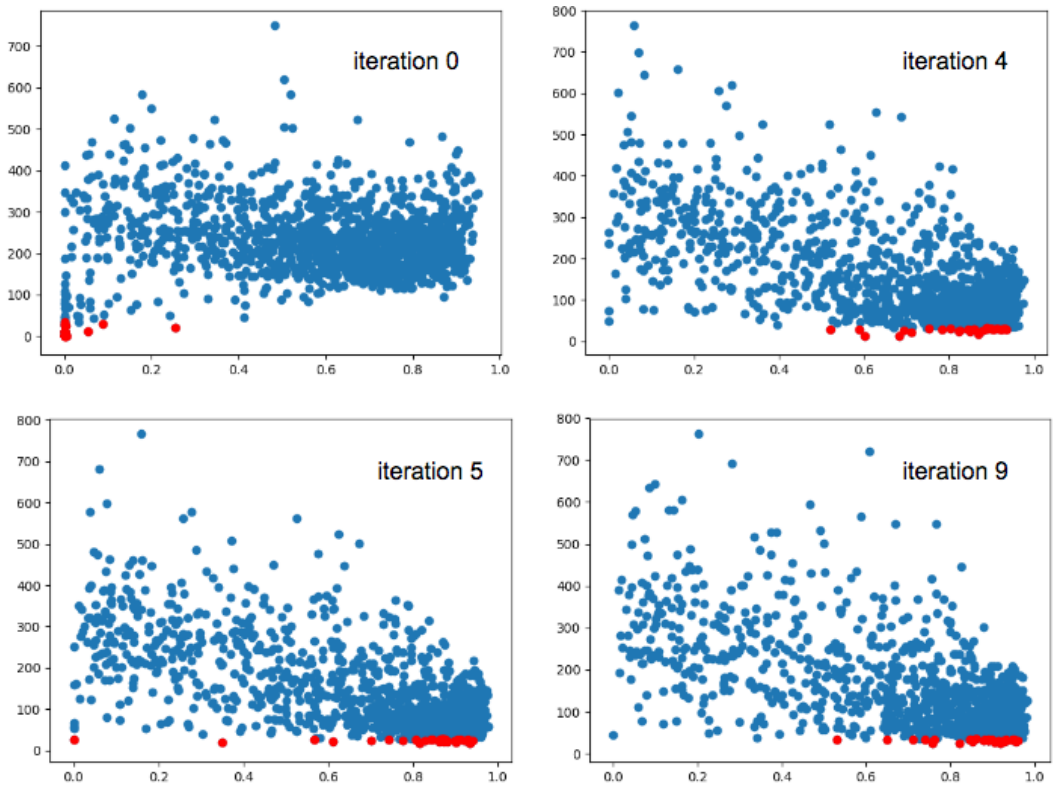


Figure 5.10: Regions digram for experiment 1. Red samples are the choses to oracle labeling

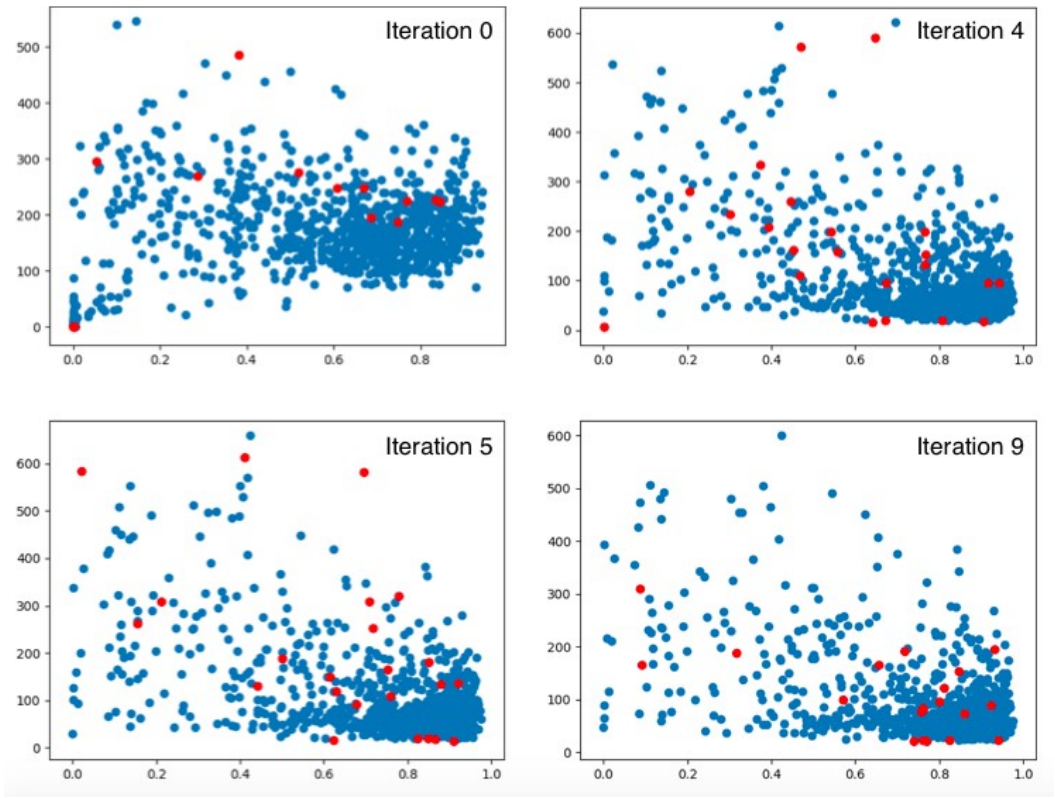


Figure 5.11: Regions digram for experiment 2. Red samples are the choses to oracle labeling. Note the improvement after epoch 5 by the increment of random samples K_4.

5.8.5 General evaluation

In order to evaluate the Active Learning performance in general terms, we used the results from the ISBI 2016 Challenge [2] for the training of the U-Net with the ISIC dataset. Each participant were ranked by several evaluation index, in our case, we based only on the Dice coefficient index, the metric used in this project. The winner of the Challenge, *Urko Sanchez* [3], obtained a Dice index of 0.90 over the 28 participants on the segmentation task. This model were trained with 300 training epochs using the overall ISIC training set of 2000 labeled samples. The best proposed model for Active Learning approach achieved a Dice index of 0.78 using 10 initial epochs and 10 interactive iterations totaling 30 training epochs. The total of labeled data between the initial labeled set and all the annotations was 900 samples.

	Classical Method	**Active Learning Method**
Best Dice Index	0,95	0,78
Training epochs	300	30
Total Labeled Data	2000	900

Table 5.4: General evaluation based on ISBI 2016 Challenge.

A visual evaluation may be interesting to made a qualitative idea of the system performance. In Figure 5.12 there are examples of what are considered satisfactory and poor segmentations.

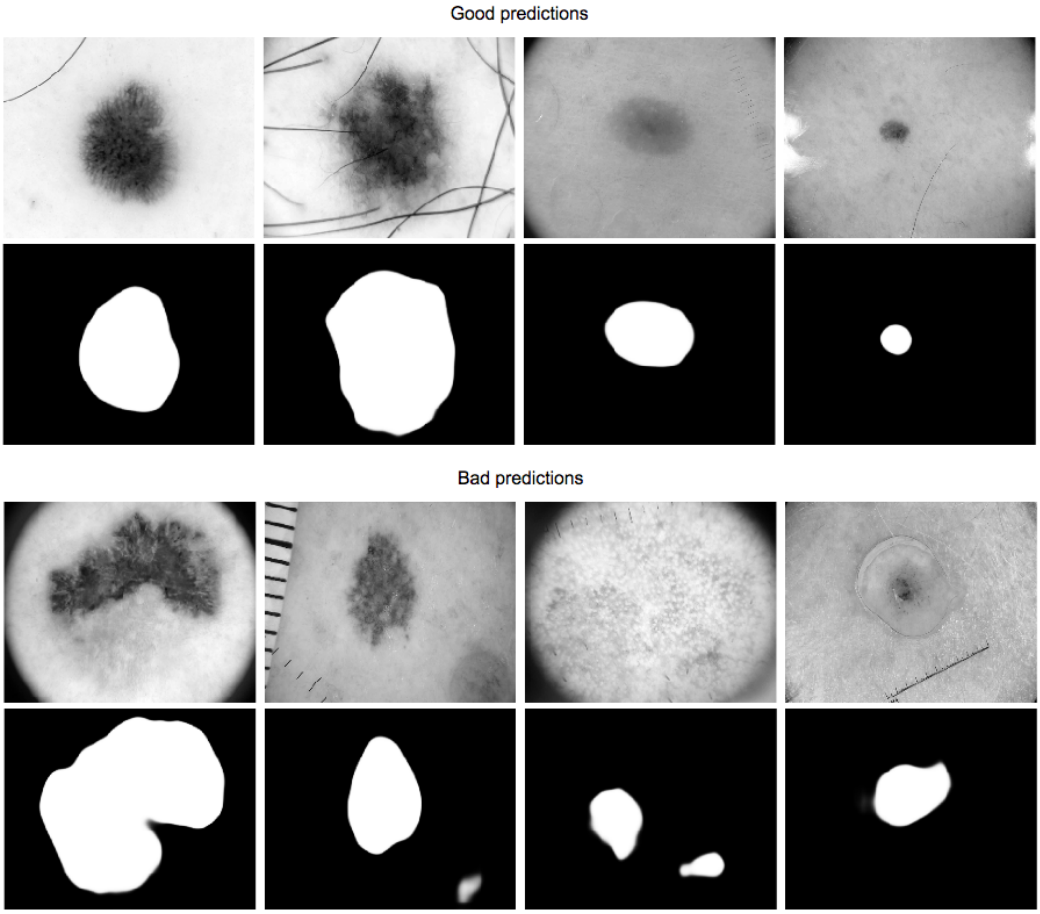


Figure 5.12: Examples of satisfactory and poor segmentation results.

Chapter 6

Budget

This project has been developed using the resources provided by Image Processing Group of UPC, and as it is a comparative study, there are not maintenance costs.

Therefore, the main costs of this projects comes from the salary of the researches and the time spent in it. I consider that my position has been as junior engineer, while the three professors who were advising me had a wage/hour of a senior engineer. I will consider that the total duration of the project was of 22 weeks, as depicted in the Gantt diagram in Figure 1.2.

	Amount	**Wage/hour**	**Dedication**	**Total**
Junior engineer	1	12,00 €/h	30 h/week	7,920 €
Senior engineer	3	20,00 €/h	4 h/week	5,280 €
			Total	13,200 €

Table 6.1: Budget of the project

Chapter 7

Appendices

7.1 Variables

D^L	*Labeled set size*
D^U	*Unlabeled set size*
D^H	*Pseudo candidates set size*
K_C	*Hight confidence selection size*
K_U	*Low confidence selection size*
K_1	*No-detections selection size*
K_2	*Most uncertain selection size*
K_4	*Central random selection size*
T	*Dropout step predictions*
W	*ConvNet parameters*
U	*Overall uncertainty metric*
U_{PW}	*Pixel-wise uncertainty map*

7.2 Definition of hyperparameters

During all the methodology, some parameters are called to be altered in order to get the best system performance. They are defined as follows:

- **Batch size**. The *batch size* is attributed to the number of training images in one forward or backward pass. It is important to highlight that the higher the batch size, the more memory will be needed.
- **Epoch**. The number of epochs measures how many times every image has been seen during training (i.e. one epoch means that every image has been seen once). It can be also understood as a one forward pass and one backward pass of all the training examples. It is numerically computed as:
- **Iterations**. The number of interactions in the Active Learning loop. Each one contains a training step with a certain defined number of epochs.
- **Loss function**. Loss function (also called cost function) evaluates the penalty between the prediction and the ground truth label in every batch.
- **Learning rate**. The learning rate parameter defines the step size for which the weights of a model are updated regarding the stochastic gradient descent. Decay. The weight decay is an additional weight update parameter that induces the weights to exponentially decay to zero once the update process is over.
- **Optimizer**. *Keras* framework provides optimizers in order to find the most optimal set of hyperparameters for the model. Some examples are the *SGD*, *RMSprop* and *Adam*.

Chapter 8

Conclusions

The main objective of this project was to propose a framework able to manage active deep learning for medical imaging segmentation. The results obtained by the ISIC 2017 Challenge dataset for Skin Lesion Analysis towards melanoma detection are satisfactory in terms to evaluate the potential of the methodology, but not enough to be compared to the classical training model. Taking into account the complexity in segmentation to face the training of deep neuronal networks from scratch with few data, this thesis proposed strategies to study the nature of the data to prevent the learning weights to fall into local minimums.

Cost-Effective Active Learning introduced the idea of automatic annotations to able the system to generate automatically labeled data increasing the amount of training data. Seeing their potential in classification we tried to follow the idea in segmentation but it had been very difficult to filter the best instances to the others knowing the importance of the small details in the learning process, and the negative impact to consider bad predictions as true labels. Several solutions were proposed to solve the problem and to prepare the best possible environment before start the process, and the results are satisfactory in terms of data analysis. The called *regions diagrams* allowed us to evaluate through a visual way the performance of the interactive methodology by seeing the effects of the most informative annotations. Although the results are not comparable than the classical training, in terms of prediction goodness, there remains an open a door for further works to keep researching new solutions for data analysis to find the instances with big impact for the overall outcome to prove if the expected potential of the pseudo labels can give the system a competitive performance.

Finally, as future work, it may be interesting to test new methods in the state-of-the-art for the complementary data selection, such as the pull the plug? [7], to achieve more correlation between the informativeness and the system performance. Another possible modifications would be related to the regions digram criteria, using metrics such as the overall uncertain variance to adapt the regions frontiers to the data in real time. As a last proposal could be great to use transfer learning by using pre-trained networks in order to initialize the weights safely.

At last, all the code from this thesis to check and reproduce all the results, is publicly available on: `https://github.com/marc-gorriz/CEAL-Medical-Image-Segmentation`

Bibliography

[1] Ieee international symposium on biomedical imaging. cell tracking challenge. [online]. 2015. http://biomedicalimaging.org/2015/hardi-reconstruction-challenge/.

[2] International symposium on biomedical imaging. isbi 2016: Skin lesion analysis towards melanoma detection. [online]. 2016. https://challenge.kitware.com/#submission/56fe2b60cad3a55ecee8cf74,.

[3] U. sanchez. isbi 2016 challenge results [online]. 2016. https://challenge.kitware.com/#challenge/560d7856cad3a57cfde481ba/,.

[4] Isic archive. international skin imaging collaboration: Melanoma project website. [online]. 2017. http://isic-archive.com/.

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

[6] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *Computer Vision–ECCV*, page 584–599, Springer, 2014.

[7] Margrit Betke Danna Gurari, Suyog Jain and Kristen Grauman. Pull the plug? predicting if computers or humans should segment images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 382–391, 2016.

[8] Devi Parik h Jiebo Luo Dhruv Batra, Adarsh Kowdle and Tsuhan Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *EEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[9] Y. Gal and Z. Ghahramani. Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference. 2015.

[10] A. Graves. Practical variational inference for neural networks. In *In Advances in Neural Information Processing Systems*, pages 2348–2356, 2011.

[11] D. Kirkpatrick M. Werman H. Breu, J. Gil. Linear time euclidean distance transform algorithms. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 17, pages 529–533, 1995.

[12] Chuan Zhou Yi-Ta Wu Berkman Sahiner Lubomir M. Hadjiiski Marilyn A. Roubidoux Jun Wei, Heang-Ping Chan and Mark A. Helvie. Computer-aided detection of breast masses: Four-view strategy for screening mammography. In *Med Phys, 38*, page 1867–1876, 2011.

[13] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. volume abs/1511.02680, 2015.

[14] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *CoRR*, abs/1210.5644, 2012.

[15] Heeyoung Kwon, Kiwon Yun, Minh Hoai, and Dimitris Samaras. The ignorant led by the blind: A hybrid human–machine vision system for fine-grained categorization. In *Int J Comput Vis*, page 3–29, 2014.

[16] Tsoi AC Back AD Lawrence S, Giles CL. Face recognition: a convolutional neural-network approach. In *IEEE Trans Neural Netw, 8*, pages 98–113, 1997.

[17] Bottou L. Bengio Y. LeCun, Y. and P Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE, 86*, page 2278–2324, 2014.

[18] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12, ACM, 1994.

[19] Haikel AlHichri Naif Alajlan Farid Melgani R.R. Yager M.M. Al Rahhal, Yakoub Bazi. Deep learning approach for active classification of electrocardiogram signals. In *Information Sciences*, volume 345, pages 340–35, 2016.

[20] E. Mohedano. Deep learning for computer vision barcelona - image classification. [online]. 2016. `http://www.slideshare.net/xavigiro/image-classification-dlcv-d1l2/`.

[21] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. *CoRR*, abs/1505.04366, 2015.

[22] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. *CoRR*, abs/1505.04366, 2015.

[23] P. Fischer O. Ronneberger and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *In International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.

[24] Xiaolong Wang Shusen Zhou, Qingcai Chen. Active deep learning method for semi-supervised sentiment classification. In *Neurocomputing*, volume 120, pages 536–546, 2013.

[25] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. volume abs/1701.03551, 2017.

[26] X.Li and Y.Guo. Adaptive active learning for image classification. In *CVPR*, 2013.