# Convolutional Neural Networks for Multi-Label Image Classification

by

Jan André Marais

*Thesis presented in partial fulfilment of the requirements for the degree of Master of Commerce (Mathematical Statistics) in the Faculty of Economic and Management Sciences at Stellenbosch University*

Supervisor: Dr. S. Bierman

December 2017

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

# Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: ................................

# Abstract

## Convolutional Neural Networks for Multi-Label Image Classification

J. A. Marais

Thesis: MCom (Mathematical Statistics)

December 2017

English abstract.

# Uittreksel

## Konvolusionele Neurale Netwerke vir Multi-Etikel Beeldklassifikasie

*("Convolutional Neural Networks for Multi-Label Image Classification")*

J. A. Marais

Tesis: MCom (Wiskundige Statistiek)

Desember 2017

Afrikaans abstract

# Acknowledgements

I would like to express my sincere gratitude to the following people and organisations ...

# Contents

**Declaration** i

**Abstract** ii

**Uittreksel** iii

**Acknowledgements** iv

**Contents** v

**List of Figures** vii

**List of Tables** viii

**List of Abbreviations and/or Acronyms** ix

**Nomenclature** xi

**1 Experiments and Results** 1

1.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

1.2 Evaluation of Approaches? . . . . . . . . . . . . . . . . . . . . . 3

1.2.1 Evaluatution Metrics . . . . . . . . . . . . . . . . . . 3

1.2.2 Validation Approach . . . . . . . . . . . . . . . . . . 4

1.3 Training Procedure . . . . . . . . . . . . . . . . . . . . . . . . 4

1.3.1 How will the policy parameters be chosen? . . . . . . . 4

1.3.2 Fine-Tuning or Global Tuning . . . . . . . . . . . . . 4

1.4 Model Architecture . . . . . . . . . . . . . . . . . . . . . . . . 5

1.5 Base Experiments . . . . . . . . . . . . . . . . . . . . . . . . . 5

1.5.1 Validation Split Experiment . . . . . . . . . . . . . . . 5

1.5.2 Architecture Experiment . . . . . . . . . . . . . . . . . 5

1.5.3 Transfer Learning Experiment . . . . . . . . . . . . . . . 5
1.6 Multi-Label Experiments . . . . . . . . . . . . . . . . . . . . . . 5
1.6.1 Loss Function Experiment . . . . . . . . . . . . . . . . . . 5
1.6.2 Classification Head Experiment . . . . . . . . . . . . . 6
1.7 Thresholding Experiment . . . . . . . . . . . . . . . . . . . . . . 6
1.8 Other Experiments . . . . . . . . . . . . . . . . . . . . . . . . . . 6
1.8.1 Spatial Pyramid Pooling . . . . . . . . . . . . . . . . . . . 6
1.9 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

**Appendices** **7**

**Bibliography** **8**

# List of Figures

# List of Tables

# List of Abbreviations and/or Acronyms

| | |
|---|---|
| **AA** | Algorithm Adaptation |
| **ANN** | Artificial Neural Network |
| **BR** | Binary Relevance |
| **CAD** | Computer Aided Diagnosis |
| **CC** | Classifier Chains |
| **CNN** | Convolutional Neural Network |
| **CV** | Computer Vision |
| **ECC** | Ensemble Classifier Chains |
| **kNN** | $k$-Nearest Neighbour |
| **LP** | Label Powerset |
| **mAP** | Mean Average Precision |
| **ML-kNN** | Multi-Label $k$-Nearest Neighbour |
| **MLC** | Multi-Label Classification |
| **MLIC** | Multi-Label Image Classification |
| **PT** | Problem Transformation |
| **RAkEL** | Random $k$-Labelsets |
| **SGD** | Stochastic Gradient Descent |

| | |
|---|---|
| **SotA** | State-of-the-Art |

# Nomenclature

| | |
|---|---|
| $N$ | number of observations in a dataset |
| $p$ | input dimension or the number of features for an observation |
| $K$ | number of labels in a dataset |
| $\boldsymbol{x}$ | $p$-dimensional input vector $(x_1, x_2, \ldots, x_p)^\intercal$ |
| $\lambda$ | label |
| $\mathcal{L}$ | complete set of labels in a dataset $\mathcal{L} = \{\lambda_1, \lambda_2, \ldots, \lambda_K\}$ |
| $Y$ | labelset associated with $\boldsymbol{x}$, $Y \subseteq \mathcal{L}$ |
| $\hat{Y}$ | predicted labelset associated with $\boldsymbol{x}$, $\hat{Y} \subseteq \mathcal{L}$, produced by $h(\cdot)$ |
| $\boldsymbol{y}$ | $K$-dimensional label indicator vector, $(y_1, y_2, \ldots, y_K)^\intercal$, associated with observation $\boldsymbol{x}$ |
| $(\boldsymbol{x}_i, Y_i)_{i=1}^N$ | multi-label dataset with $N$ observations |
| $D$ | dataset |
| $h(\cdot)$ | multi-label classifier $h : \mathbb{R}^p \to 2^{\mathcal{L}}$, where $h(\boldsymbol{x})$ returns the set of labels for $\boldsymbol{x}$ |
| $\theta$ | set of parameters for $h(\cdot)$ |
| $\hat{\theta}$ | set of parameters for $h(\cdot)$ that optimise the loss function |
| $L(\cdot, \cdot)$ | loss function between predicted and true labels |
| $f(\cdot)$ | label prediction module, $f : \mathbb{R}^p \to \mathbb{R}^K$ |
| $t(\cdot)$ | thresholding function, $t : \mathbb{R}^K \to \{0, 1\}^K$ |
| $\mathcal{N}(\boldsymbol{x})$ | points in the input space neighbourhood of $\boldsymbol{x}$ |

# Chapter 1

# Experiments and Results

*"For us, the most important part of rigor is better empiricism, not more mathematical theories."*

— Ali Rahimi and Ben Recht, *NIPS 2017*

## 1.1 Introduction

We want all of this experiments to be as reproducible as possible.

Old introduction:

> The main aim of this chapter is to empirically compare some of the deep learning for multi-label image classification approaches proposed in the literature in a standardised fashion. We will also attempt to empirically answer some of the questions that arose in the literature study.
>
> Multi-label image classification with CNNs is still a relatively new research area. No work has been done to provide an extensive and robust comparison of the existing approaches in the literature. Typically, when a new approach is proposed it is empirically compared to other previous proposed approaches. But these evaluations of the approaches are not in a standardised fashion. The base networks and optimisation procedures are just some of the learning components that vary accross the proposed approaches. This makes it difficult to determine whether or not a proposed approach performs empirically better than another because of its

ability to model multi-label images or because of the latest general developments of training CNNs.

Take the Spatial Regularisation Network (SRN) in the previous chapter as an example. The SRN is an extension of a base CNN that is supposed to help exploiting spatial relations amongs labels. The SRN shows favourable empirical results over all other proposed approaches. However, it also uses a much deeper base CNN (ResNet-101) than the other approaches in the literature. This makes it difficult to determine whether or not the performance boost comes from the SRN or the deeper CNN.

For this reason, we want to provide a standardised and robust comparison of the some the most promising approaches in the literature. To standardise the comparisons, we will evaluate the chosen approaches using the same base CNN and optimisation procedure. To ensure robustness, we will evaluate the methods on two very distinct multi-label image datasets (described in Appendix **??**), using multiple diverse evaluation metrics and using cross-validation for a better estimate of generalisation ability which will also allow us to report standard deviations of errors.

There are 4 main question we attempt to answer in this chapter. They are:

1. How do the different loss functions act as a surrogate for the micro and macro F-score? (bce vs weighted bce vs rank loss vs retina loss)

2. Does multi-level predictions help to detect small objects?

3. Which extension works best to explicitly model label correlations? CG vs chaining vs SE-module

4. How does learnable label calibration modules compare to brute force search?

> After getting closer to the answers of these questions we will train a final model taking into considerations the empirical findings to see how accurate we can get on both datasets.

End of old introduction.

## 1.2 Evaluation of Approaches?

### 1.2.1 Evaluatution Metrics

We chose the following metrics to measure the performance of the model on the data:

- Label-based macro $F_1$-score ($F_1^{\text{macro}}$),
- Label-based micro $F_1$-score ($F_1^{\text{micro}}$),
- example-based average precision (AP), and
- Label-based macro ROC-AUC

By using these four metrics we will get an all-round estimate of the performance the models. This is a diverse set of metrics. Includes label-based and example-based metrics, $F$-score, AP and ROC-AUC metrics and micro- and macro-average metrics. The $F$-score metric variants are popular choices for evaluating MLC models. The AP metric is common in the Computer Vison domain. The ROC-AUC is chosen mainly to be able to compare the models to other work reported on this dataset. ROC-AUC is also a convenient option since it is independent of the classification threshold chosen. When applicable, we will inspect the performance of the models on a per label basis.

For the $F$-scores, we will need to threshold the outputs coming from the CNN. We will not search for the optimal threshold for every experiment. Rather use a standard threshold of 0.5 when comparing between models. BUt if we want to squeeze out more accuracy, we will search for the optimal threshold.

When possible, the chosen set of metrics will be reported after each epoch in the form of line graphs. We do it this way because the point of convergence for the loss function being trained on might not be the same as the metric reported. Thus, if we only report the performance of the final (converged) model, we might not see the best possible performance for each of the metrics.

The performance of the best (and/or final - I must still choose) models for each training phase will be reported in tabular form.

The final model evaluations will be reported on both the validation and testing sets. No model selection will be done on the test set evaluations.

Where possible we will include the time taken to train until convergence. Also time taken to make a prediciton for a single image.

### 1.2.2 Validation Approach

The data is split into a training, validation and test set. Since our dataset is large and our computing resources limited, we are comfortable not to use cross-validation. We can do a cross-validation of the final model to have a better estimate of it's performance. We will use the exact same split as in (paper) for fairer comparisons. The split was made randomly by patient in the following ratios: 70% training, 10% validation, 20% testing. There is no overlap between the patients in different splits to ensure uniqueness of the validation and testing examples.

## 1.3 Training Procedure

1Cycle policy with learning rate finder. Adam optimiser.

### 1.3.1 How will the policy parameters be chosen?

See paper. Parameters:

- ratio between minimum and maximum learning rate,
- decay rate
- momentum decay
- weight decay?

### 1.3.2 Fine-Tuning or Global Tuning

Prefer fine-tuning where possible to save time. Will not be as accurate as global tuning. Can precompute the activations before the classification head to be tuned, which saves a lot of repetitive computing. Will do complete global tuning for specific models where appropriate. Will use appropriate data augmentation techniques when precomputed activations are not used. Give a more detailed explanation.

## 1.4 Model Architecture

Will have an initial experiment to compare different architectures. The majority of the experiments we will run using the smallest version of the chosen architecture type to reduce computational demand. We assume the conclusions are applicable to the larger models unless specified otherwise.

Sometime we will use smaller versions of the x-ray, say 128x128 to make the experiments run faster. Which again we assume the same conclusions will hold for the original x-ray sizes.

## 1.5 Base Experiments

### 1.5.1 Validation Split Experiment

Will it bias results to split data randomly?

### 1.5.2 Architecture Experiment

Which of the following architectures perform the best on our data ResNets, DenseNets, SEResNet, DarkNet?

### 1.5.3 Transfer Learning Experiment

Does it help to do transfer learning vs training from scratch? Does it help to pretrain on another x-ray based dataset?

## 1.6 Multi-Label Experiments

### 1.6.1 Loss Function Experiment

By training on which loss function will result in the best metrics?

The goal of this experiment is to find out which multi-label loss function is a more suitable surrogate for our chosen multi-label evaluation metrics. We compare the binary cross-entropy loss, weighted binary cross-entropy loss, focal loss and LSEP loss. We only use one ranking based loss function since there was significant proof given in (Li *et al.*, 2017) that LSEP loss outperforms the other ranking based loss functions. We will also experiment with the focal loss weighted as in W-CE. This is the first time the focal loss is used with multi-label image classification and the first time LSEP loss is compared to cross-entropy based loss functions.

### 1.6.2 Classification Head Experiment

Which classification head architecture obtains the best results? Does it learn label dependence?

## 1.7 Thresholding Experiment

Can the network learn the optimal threshold?

## 1.8 Other Experiments

### 1.8.1 Spatial Pyramid Pooling

Does it help to make predictions from multiple layers of the CNN?

## 1.9 Summary

Write summary here.

# Appendices

# Bibliography

Li, Y., Song, Y. and Luo, J. (2017). Improving pairwise ranking for multi-label image classification. *CoRR*, vol. abs/1704.03135. 1704.03135.
Available at: http://arxiv.org/abs/1704.03135