The 2018 Medico Multimedia Task Submission of Team NOAT using Neural Network Features and Search-based Classification

Michael Steiner[1], Mathias Lux[1], and Pål Halvorsen[2,3]
[1]Alpen-Adria-Universität Klagenfurt, Austria; [2]SimulaMet, Norway; [3]University of Oslo, Norway;
michstei@edu.aau.at,mlux@itec.aau.at,paalh@simula.no

ABSTRACT

In this paper, we describe our approach for the classification of medical images depicting the human gastrointestinal tract. Search-based classification is performed in three stages. In the first stage, we extract deep features for each image using pre-trained deep-learning models [1]. In the second stage, we use LIRE [3] to index the generated features, so that we can then, in the final stage, search the index for similar images and make our predictions based on the results. With this approach, we achieved a MCC score of $0,54$ and a accuracy of $0,94$, which shows that deep features combined with search-based classification are a viable option for medical image analysis.

1 INTRODUCTION

The aim of the 2018 MEDICO task [5] is to classify images of the gastrointestinal (GI) tract into the provided categories. The task provides images and several pre-extracted global image features. The images of the development set [4] are categorized into 16 classes. In our approach, we use the extracted features of pre-trained deep-learning models [1] to perform a search-based classification [6] using LIRE [3].

2 FEATURE EXTRACTION

The deep-learning features are extracted with a Python script, using the Keras API [1]. For this task we used features from the following models: DenseNet121, DenseNet169, DenseNet201, ResNet50, MobileNet, VGG16, VGG19, Xception.

The models are pre-trained on the ImageNet dataset [2] and chosen based on their result on the ImageNet dataset. For the VGG16 and VGG19 models, we are not using the four top layers, which are used for the ImageNet predictions, and we use a GlobalMaxPooling2D layer as final layer. This leaves us with feature vectors of 512 values for these two models.

For the other models we are also not using the top layers, but in addition to the GlobalMaxPooling2D layer we also added a Dense layer to get a length of 1024 values for our feature vectors. The generated feature vectors are then stored in comma-separated-values (CSV) files. We create one file for each model and it contains the filename followed by the feature vector. Before the indexing of the files we also perform a quantization step to bring the feature vectors from double range to integer range. This step not only increased the accuracy in our tests, but it also made the index smaller and reduced the processing time for indexing and searching. The different models are then combined in the indexing stage. To do this we create one entry per model per file in the index, resulting in 8 index entries per file.

3 INDEXING

In order to index the extracted features (see section 2) with LIRE [3], we first created java classes for each model to represent the feature. The classes are based on the existing classes for global features, to leave the option of combining the deep-learning features with the already implemented global features open. These new classes were necessary because we need a way to retrieve the features from the CSV files rather than the images themselves. Next, we adapted LIRE's document builder class to support the new feature classes and then indexed the images using the pre-computed feature vectors with locality sensitive hashing (bit sampling) and/or metric spaces hashing.

4 SEARCHING

Analog to the indexing in section 3, we had to adapt the existing LIRE classes for searching the index to support the new feature classes and work with file paths instead of images. To find most similar images, we used the cosine distance function, which was already implemented in LIRE.

5 CLASSIFICATION

For the classification, we took the best nine results of each model and generated predictions for each model by counting the returned categories. These intermediate predictions are weighted and combined to form the final prediction. The weights are manually generated based on experiments on the development data set [4].

6 RESULTS

For our submitted runs we used all the files of the training dataset to create the index. We used following configurations for our four submitted runs:

- Run 1: Integer Features with Bitsampling Hashing
  The extracted features are quantized, to use integer values instead of double values, and then indexed using bitsampling hashing.
- Run 2: Integer Features with Metric Spaces Hashing
  The extracted features are quantized, to use integer values instead of double values, and then indexed using metric spaces hashing.
- Run 3: Integer Features with Bitsampling and Metric Spaces Hashing
  The extracted features are quantized, to use integer values instead of double values. Then they are indexed using

MediaEval'18, 29-31 October 2018, Sophia Antipolis, France

Table 1: Official run submission results

	Recall	Specificity	Precision	Accuracy	F1	MCC	R_k statistic
Run 1	0.5756	0.9717	0.5756	0.9469	0.5756	0.5473	0.5368
Run 2	0.3677	0.9578	0.3677	0.9209	0.3677	0.3255	0.3194
Run 3	0.5371	0.9691	0.5371	0.9421	0.5371	0.5063	0.5039
Run 4	0.5667	0.9711	0.5667	0.9458	0.5667	0.5378	0.5282

Table 2: Confusion Matrix for Run 1.
(Classes: blurry-nothing (BLN), colon-clear (COC), dyed-lifted-polyps (DLP), dyed-resection-margins (DRM), esophagitis (ESO), instruments (INS), normal-cecum (NOC), normal-pylorus (NOP), normal-z-line (NZL), out-of-patient (OOP), polyps (POL), retroflex-rectum (RER), retroflex-stomach (RES), stool-inclusions (STI), stool-plenty (STP), ulcerative-colitis (ULC))

Pred \Act class	ULC	ESO	NZL	DLP	DRM	OOP	NOP	STI	STP	BLN	POL	NOC	COC	RER	RES	INS
ULC	366	15	5	53	51	0	44	0	23	0	53	7	4	53	38	40
ESO	0	1	0	2	4	0	5	0	1	0	2	0	0	1	2	2
NZL	2	275	485	4	1	1	51	3	3	0	4	0	7	3	17	4
DLP	55	52	26	241	169	0	130	98	123	0	148	37	122	61	180	96
DRM	48	83	24	163	252	3	114	36	62	14	64	12	40	35	66	84
OOP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NOP	1	1	0	3	1	0	38	0	0	0	7	0	0	0	2	2
STI	0	0	0	0	0	0	0	365	0	0	0	0	0	0	0	1
STP	3	6	0	2	2	0	4	0	1732	0	1	0	0	0	1	2
BLN	0	0	0	11	12	0	0	0	2	23	0	0	2	0	1	3
POL	21	5	2	30	26	0	89	2	8	0	41	2	2	27	22	25
NOC	43	0	0	40	40	1	2	0	2	0	45	526	0	5	0	8
COC	0	0	0	0	0	0	0	1	5	0	0	0	888	0	0	0
RER	0	0	0	1	0	0	0	0	0	0	0	0	0	4	1	1
RES	3	118	21	6	6	0	84	1	4	0	9	0	0	3	67	3
INS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2

bitsampling hashing and metric spaces hashing, resulting in two indexes that are searched for the classification.

- Run 4: Double Features with Bitsampling Hashing
  The extracted features are not quantized and then indexed using bitsampling hashing.

As shown in Table 1, we achieved the best results with Run 1. Here, we used the features from the deep learning Models, configured as described in Section 2, and quantized them from the double range to integer range. Run 2 shows, that metric spaces hashing seems to be less suited for this task. It has given better result in predicting Esophargitis correctly, but over all the results are significantly worse. And in the combination of both hashing methods (run 3), we could notice that for example the correct prediction of Esophargitis got slightly better, but overall the worse performance of metric spaces had a negative impact on the prediction. Run 4 shows that the quantization from double to int brings a slight increase in overall performance.

Table 2 shows the confusion matrix of our best results in Run 1. The main problem areas here are the classification of Esophargitis as Normal-Z-Line and mixing up Dyed-Resection-Margins with Dyed-Lifted-Polyps. The results for the classes Out-Of-Patient (OOP) and Instruments (INS) are very low, this is mostly because of the lack of example images in the dataset [5]. With only four images for OOP and 36 for INS, it is almost impossible to find similar images, especially since images in these two classes can vary a lot more compared to the other classes.

7 DISCUSSION & OUTLOOK

In this paper, we showed an approach to utilize deep feature vectors for search-based classification, by extracting the features with deep neural networks and indexing those features to make them searchable. We got an accuracy of $0,94$ and MCC score of $0,54$. This was achieved by using bitsampling indexing combined with features quantized to the integer range, which may cause a noise reduction compared to the double features. Further experiments could be made, to see if better results can be achieved by training the models on medical images rather than using the models trained on the ImageNet dataset[2].

ACKNOWLEDGEMENTS

We would like to thank our colleagues *who helped* for all the input and discussions. Travel was funded by Alpen-Adria Universität Klagenfurt.

REFERENCES

[1] François Chollet and others. 2015. Keras. https://keras.io. (2015).

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

[3] Mathias Lux and Savvas A. Chatzichristofis. 2008. Lire: Lucene Image Retrieval: An Extensible Java CBIR Library. In *Proceedings of the 16th ACM International Conference on Multimedia (MM '08)*. ACM, 1085–1088. https://doi.org/10.1145/1459359.1459577

[4] Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, and Pål Halvorsen. 2017. Kvasir: A Multi-Class Image Dataset for Computer Aided Gastrointestinal Disease Detection. In *Proceedings of the 8th ACM on Multimedia Systems Conference*. ACM, 164–169.

[5] Konstantin Pogorelov, Michael Riegler, Thomas De Lange, Kristin Ranheim Randel, Mathias Lux, and Olga Ostroukhova. 2018. Medico Multimedia Task at MediaEval 2018. 7 (2018), 7–9.

[6] Michael Riegler, Martha Larson, Mathias Lux, and Christoph Kofler. 2014. How 'How' Reflects What's What: Content-based Exploitation of How Users Frame Social Images. In *Proceedings of the 22Nd ACM International Conference on Multimedia (MM '14)*. ACM, New York, NY, USA, 397–406. https://doi.org/10.1145/2647868.2654894