Feature Learning with Adversarial Networks for Concept Detection in Medical Images: UA.PT Bioinformatics at ImageCLEF 2018

Eduardo Pinho and Carlos Costa

DETI - Institute of Electronics and Informatics Engineering of Aveiro
University of Aveiro, Portugal
{eduardopinho, carlos.costa}@ua.pt

Abstract. As the subjects of representation learning and generative adversarial networks become increasingly attractive to the scientific community, they also bring an exciting perspective towards their application in the digital medical imaging domain. In particular, the ImageCLEF caption challenge is focused on automatic medical image understanding. This paper describes a set of feature learning approaches for the concept detection sub-task of ImageCLEFcaption 2018. The first approach consists on a traditional bag of words algorithm, using ORB keypoint descriptors. The remaining two methods are based on a variant of generative adversarial networks with an auto-encoding process. Subsequently, two kinds of classification algorithms were employed for concept detection over the feature spaces learned. Test results showed a best mean F1 score of 0.110176 for linear classifiers, by using the features of the adversarial autoencoder.

Keywords: ImageCLEF · Representation Learning · Deep Learning · Generative Adversarial Networks · Auto-Encoders

1 Introduction

The ImageCLEF initiative [1] has launched the second edition of the **caption** challenge [2], aiming for the automatic information extraction from medical images. This challenge is divided into two sub-tasks: *concept detection* and *caption prediction*. The *concept detection* sub-task is the first part of the caption prediction task, in which the goal is to automatically recognize certain concepts from the UMLS vocabulary [3] in biomedical images. The obtained list of concepts would then be used in the *caption prediction* sub-task, where a small human-readable description of the image is generated.

For this challenge, we reiterate on the lessons learned from the ImageCLEF 2017 concept detection task [4], by experimenting with new representation learning methods, through which other machine learning tasks can be employed more efficiently, including biomedical concept detection. This is also part of a wider vision

of improving a system's information retrieval capabilities over non-annotated data.

This paper presents our solution proposal for the concept detection sub-task, and describes our methods of image feature extraction for the purpose of biomedical concept recognition, followed by their evaluation under the ImageCLEF 2018 challenge.

2 Materials and Methods

This task was accompanied with two data sets containing various images from biomedical literature: the *training set*, comprising 223,859 images, included the list of concepts from the UMLS dictionary associated to each image. The *testing set*, composed of 9938 images, had its annotations hidden from the participants.

We have addressed the concept detection task in two phases. First, mid-level representations of the images were chosen and built:

- In Section 2.1, as a classical approach, bags of visual words were used as image descriptors, obtained from the clustering of visual keypoints;
- In Section 2.2, two kinds of deep neural networks for unsupervised feature learning were designed and trained for the purpose of visual feature extraction.

Afterwards, as described in Section 2.4, the concept detection problem were treated as a multi-label classification problem: the three representations were validated by training classifiers of low complexity over the new representations.

2.1 Bags of Visual Words

After converting the images to grayscale, without resizing, visual keypoint descriptors were extracted using Oriented FAST and Rotated BRIEF (ORB) [5]. The implementation in OpenCV [6] was used for ORB keypoint extraction and descriptor computation. Each image would yield a variable number of descriptors of size 64. In cases where the ORB algorithm did not retrieve any keypoints, the algorithm's parameters were adjusted to loosen edge detection criteria. As a last resort, images without any meaningful features (e.g. images of solid or gradiented color) were given an empty bag of words.

From the training set, five thousand files were randomly chosen and their respective keypoints collected to serve as a template keypoint set. A visual vocabulary (codebook) of size $k = 4096$ was then obtained by performing k-means clustering on all template keypoints and retrieving the centroids of each cluster, yielding a list $\mathcal{V} = \{V_i\}$, of 4096 64-dimensional vectors. We used the k-means clustering implementation from the Faiss library [7].

Once a visual vocabulary was available, each image's bag of visual words (BoW) was constructed by determining the closest visual vocabulary point and incrementing the corresponding position in the BoW for each image keypoint descriptor. In other words, for an image's BoW $B = \{o_i\}$, for each image keypoint descriptor d_j, o_i is incremented when the smallest Euclidean distance from d_j to all other visual vocabulary points in V is the distance to V_i. Finally, each BoW was normalized so that the maximum value of the elements in each BoW equals 1. The visual BoWs method has been widely experimented in content based image retrieval (CBIR) for representing visual content, including medical images [8]. From our prior experience in visual BoWs, we deem ORB as a competitive, open, and faster alternative to more frequently employed keypoint detection algorithms, such as Scale Invariant Feature Transform (SIFT) [9].

2.2 Adversarial Auto-encoding networks for feature learning

Generative adversarial networks (GANs) [10] are a strong target of research nowadays. These adversarial networks are primarily composed of a *generator* which learns to produce samples from a distribution, and a *discriminator* that learns to distinguish real samples from generated ones. Well trained GANs for images can produce visually appealing samples, sometimes unrecognizable from real content. This breakthrough led the scientific community into devising new GAN variants and applications to this adversarial loss, including for unsupervised representation learning [11].

One of the shortcomings of the traditional GAN schematic is the lack of a function mapping the sample distribution to the code feature space. One of the possible solutions in literature to this issue is the expansion of the GAN to hybrid architectures with an auto-encoding procedure. The following sections describe two GAN-based deep neural network models for the unsupervised extraction of visual features from biomedical images. Both architectures will result in an *encoder* of samples into a latent code z, which is subsequently used as a global descriptor.

2.2.1 Encoder / Decoder Specification The networks presented here abide to similar specifications: encoders and discriminators were built according to Table 1. The *code discriminator* is an exception to these two forms, and is instead specified and explained in Section 2.2.2. Both architectures are composed of a sequence of blocks of convolutional layers, where each are followed by a normalization procedure and leaky Rectified Linear Unit (LReLU) activations with a leakiness factor of 0.2. All convolutional layers relied on a kernel of size 3x3. The encoding network ends with a fully connected layer, where h is equal to the size of z for the encoder, and 1 for the discriminator.

Tbl. 1: A tabular representation of the sequential composition of the encoders and discriminators in the networks.

Layer	kernel size/stride	Output shape
conv($nb = 64$, LReLU)	3x3 /2	32x32x64
conv($nb = 128$, LReLU)	3x3 /1	32x32x128
conv($nb = 128$, LReLU)	3x3 /2	16x16x128
conv($nb = 256$, LReLU)	3x3 /1	16x16x256
conv($nb = 256$, LReLU)	3x3 /2	8x8x256
conv($nb = 512$, LReLU)	3x3 /1	8x8x512
conv($nb = 512$, LReLU)	3x3 /2	4x4x256
conv($nb = 512$, LReLU)	3x3 /1	4x4x512
flatten	N/A	8192
fc(nb=h)	N/A	h

The decoding network, which is used for decoders and generators, replicate the encoding process in inverse order (Table 2). It starts with a mapping of the prior (or latent) code features to a feature space of dimensions 4x4x512 using a fully connected network. Each 2-layer block is composed by a convolutional layer, followed by a transposed convolution of stride 2 (also called *fractionally-strided convolution*). Each block doubles the height and width of the output, as a consequence of the strided convolution, until the intended image output dimensions are reached. The last convolution maps these activations to the RGB pixel value domain.

Tbl. 2: A tabular representation of the sequential composition of the decoders and generators in the networks.

layer	kernel size/stride	output shape
fc($nb = 8192$, LRelU)	N/A	8192
reshape(4x4)	N/A	4x4x512
conv($nb = 512$, LRelU)	3x3 /1	4x4x512
dconv($nb = 512$, LRelU)	3x3 /2	8x8x512
conv($nb = 256$, LReLU)	3x3 /1	8x8x256
dconv($nb = 256$, LReLU)	3x3 /2	16x16x256
conv($nb = 128$, LReLU)	3x3 /1	16x16x128
dconv($nb = 128$, LReLU)	3x3 /2	32x32x128
conv($nb = 64$, LReLU)	3x3 /1	32x32x64
dconv($nb = 64$, LReLU)	3x3 /2	64x64x64
conv($nb = 32$, LReLU)	3x3 /1	64x64x64
conv($nb = 3$, tanh)	1x1 /1	64x64x3

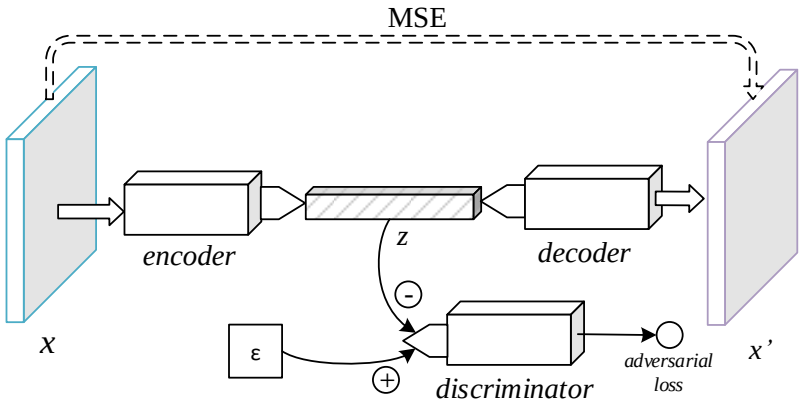

Fig. 1: Architecture of the adversarial autoencoder.

2.2.2 Adversarial Autoencoder The adversarial autoencoder makes use of the same loss function in GAN training, but to the latent vector space instead of the sample distribution [12]. As an autoencoder, its reconstruction criterion is to learn the pair of functions (E, D) so that $x' = E(G(x))$ is closest to the original sample x. In practice, this translated into a loss function for minimizing the mean squared error (MSE) between the original image and the generated one:

$$\mathcal{L}_{rec(E,G)} = \frac{1}{2N} \sum_{i}^{N} (\mathrm{x}_i - \mathrm{x'}_i)^2 \tag{1}$$

Additionally, the bottleneck vector is regularized with an additional code discriminator network. The adversarial loss function is similar to the original solution to the min-max game [10], where the value function $V(E, D)$ is solved instead: the discriminated distribution learns to distinguish the distribution $q(z \sim E(x))$ from a prior distribution $p(z)$.

$$V(E, D) = \min_{E} \max_{D} \mathbb{E}_{\mathrm{z} \sim p_{\mathrm{z}}}[\log D(\mathrm{z})] + \mathbb{E}_{\mathrm{x} \sim p(\mathrm{x})}[\log 1 - D(E(\mathrm{x}))] \tag{2}$$

Instead of convolutional layers, the code discriminator is composed of 3 fully connected networks of 1024 neurons each, with layer normalization [13] and LReLU after each one. The fourth layer, as in the remaining discriminators, ends with a single output neuron.

2.2.3 Flipped Adversarial Autoencoder When the mappings of the AAE are inverted, this results in the flipped adversarial autoencoder (F-AAE) [14]. In this architecture, a basic GAN is augmented with an encoding network $E(x') = z'$, which learns to reconstruct the original prior code leading to the given generated sample (Figure 2). As a means of stabilizing the the GAN training process, the architecture was adjusted to handle two levels of samples at different levels of the network. The generator produces two images x' and x'_{small}, the latter 4 times smaller in side (16x16), whereas the discriminator receives both images for discriminating the sample as a whole. The adversarial training formula is equivalent to the original GAN's:

$$V(G, D) = \min_G \max_D \mathbb{E}_{\mathrm{x}\sim p_{\mathrm{x}}}[\log D(\mathrm{x})] + \mathbb{E}_{\mathrm{z}\sim p(\mathrm{z})}[\log 1 - D(G(\mathrm{z}))] \tag{3}$$

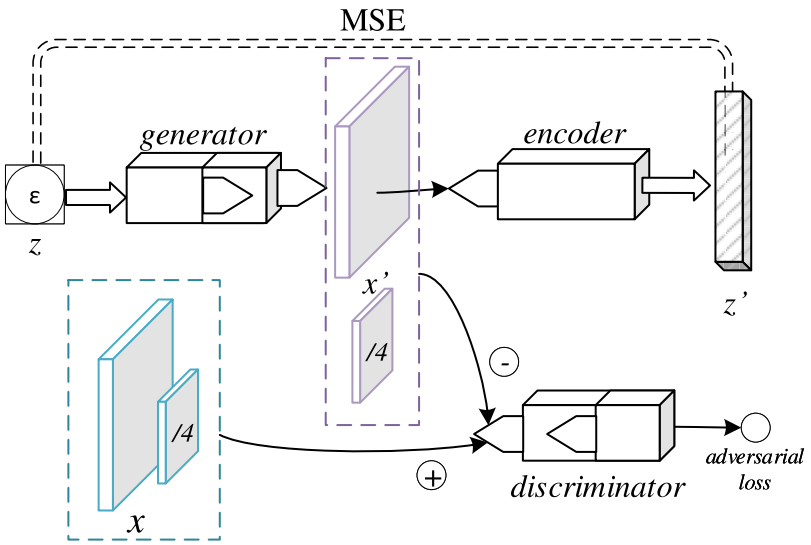

Fig. 2: Architecture of the flipped adversarial autoencoder with two image levels.

This two-level GAN is influenced by the progressive GAN [15]. After two convolutional blocks of the generator, a 1x1/1 convolution of 3 kernels is used to convert the feature maps into a smaller RGB image. The network progresses as usual to the other two blocks in order to produce the 64x64 image. At the two-level discriminator, the smaller image is mapped to a 64-channel feature map with another 1x1 convolution, and concatenated with the features after the second

convolutional block. Unlike the progressive GAN, the two levels are generated and updated simultaneously.

This idea has been reiterated in literature: the original work on GANs [10] hints towards an approximate inference component for predicting the prior code from a sample. A very similar concept was also presented by Donahue et al [16] when proposing a *latent regressor* between the prior z and an encoder network's output $E(G(z))$, while also exposing it as a major drawback in feature learning. In this architecture, the encoder never gets to see real data, and the generated samples, while potentially making great approximations of the intended distribution, will usually not achieve a perfect result. The application of the F-AAE in this task was not envisioned as a potentially competitive solution to feature learning, but rather as a means to obtain quantitative results in contrast to the AAE.

2.2.4 Image Preprocessing and Augmentation Training samples were preprocessed in the following fashion: images were resized to the square resolution of 96 pixels. Afterwards, each incoming sample was cropped to a random square of size 64x64, yielding the final *real* samples. Images in the testing set, on the other hand, were only resized to fit the 64x64 dimensions. For all cases, the images' pixel RGB values were normalized with the formula $\mathrm{n}(v) = {}^{v}/_{127.5} - 1$, thus sitting in the range [-1, 1].

2.2.5 Network Training Details The GANs were trained with sequential 3-step iterations of stochastic gradient descent: (1) reconstruction, (2) generator training, and (3) discriminator training. The prior codes were sampled from a 1024-dimensional surface of a hypersphere. The parameters were initialized with a random Gaussian distribution with a standard deviation of 0.002. Training took place through stochastic gradient descent, using the Adam optimizer [17] with a learning rate of 10^{-4} and the hyperparameters $\beta_1 = 0.5$ and $\beta_2 = 0.99$ for all three optimization steps. The AAE was trained for 140000 iterations, the F-AAE for 210000 iterations, with a mini-batch size of 32. This is approximately 20 epochs over the training data for the AAE and 30 epochs for the F-AAE.

Depending on its purpose, specific regularizers and stabilization methods were also added to the network:

- In the (convolutional) discriminator, batch normalization [18] was employed after every convolutional layer of the encoding blocks. As a technique devised in [15] to prevent mode collapse, the across-minibatch standard deviation of the last convolutional layer activations was injected into the same activations as an additional feature map, yielding a new tensor of output 4x4x513 (and a flattened layer of 8208 features). Moreover, we included an auxiliary loss component $0.001 \times \mathbb{E}[D(x)^2]$ to prevent the output from drifting too far away from zero [15].

– In the final activations of the encoder, a regularization loss was employed so that the codes would approach a unit norm, which is an invariant in the hypersphere distribution: $\epsilon_{sphere} \times |\|z\|_2 - 1|$, where ϵ_{sphere} was set to the constant 0.001.
– In the generators of samples, pixelwise feature vector normalization [15] was added immediately after every convolutional layer in the decoding blocks (before the leaky ReLU): $b_{x,y} = a_{x,y} / \sqrt{\frac{1}{N} \sum_{j=0}^{N-1} (a_{x,y}^j)^2 + \epsilon}$, where $\epsilon = 10^{-8}$.
– The introduction of some noise in GANs is known to stabilize the training process and contribute to an easier convergence [19]. We added drop-out layers [20] with a 50% drop rate at the second to last layer of the discriminator, and after the fully connected layer in the generator (25% drop rate).

TensorFlow [21] with GPU support was used to train both neural networks, as well as to retrieve the final features of each image in the two datasets.

2.3 Qualitative GAN Results

The original concept of GAN shows ground-breaking results in the field of data synthesis: as the generator learns to create realistic samples (in this case, images in the biomedical domain), retrieving new content from a trained generator can be done by feeding it with prior codes. The F-AAE benefits from this perk, as it is an extension to the original GAN. The AAE, on the other hand, presents blurrier images as a consequence of mean squared error being used as the reconstruction loss. This can be seen in Figure 3, where a few samples were retrieved nearly at the end of training.

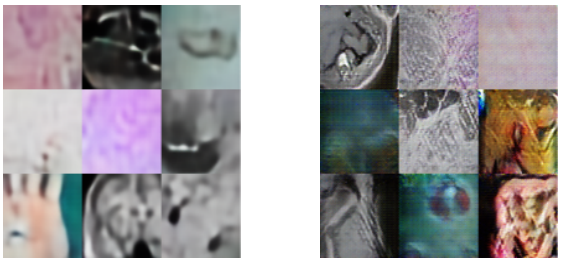

Fig. 3: An example of samples produced by the AAE and the F-AAE during the training process.

Other patterns of the learned representation can be obtained with an observation of samples in a close manifold. Figure 4 show one generated sample of the trained F-AAE which has been moved in its own latent space around one of the circles

of the hypersphere. This exploration of the feature space does not ensure that the images to stay in the same modality or retain semantic concepts, but unlike an interpolation in pixel space, intermediate positions still exhibit sharp visual features, and can be considered as "real-looking" as the starting image, within the capabilities of this GAN.

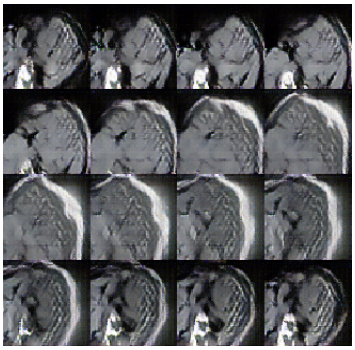

Fig. 4: A sequence of generated samples from the F-AAE by moving the prior code around a circle in the hypersphere, column-first (from left to right, then top to bottom).

2.4 Multi-label Classification

For each of the three representations learned, a multi-label classifier was applied to the resulting features, where the concepts of the image associated to the feature vector were treated as labels. With the purpose of evaluating the descriptiveness of these features, algorithms of low complexity were chosen. We experimented with logistic regression and a variant of k-nearest neighbors.

2.4.1 Logistic regression After mapping all images in the training and testing sets into the latent feature space, the training set was split to create a fine-tuning (or validation) set containing approximately 10% of the data points. Afterwards, linear classifiers were built for multi-label classification of biomedical concepts. As the number of possibly existing concepts in the images is very high, only the 500 terms most frequently occurring in the training set were considered. The label vectors were built based on a direct mapping from the UMLS term identifier to an index in the vector. The reverse mapping was kept for producing the textual list of concepts.

The linear classifiers were trained for each of the three representations, using FTRL-Proximal optimization [22] with a base learning rate of 0.05, L_1- and

L_2-norm of 0.01, and a batch size of 64. After each epoch, the classifiers were evaluated based on their precision, recall, and mean F_1 score averaged against the samples in the separate fine-tuning set, with respect to multiple fixed operating point thresholds: 0.05, 0.075, 0.1, 0.125, 0.15, 0.175, and 0.2. Classifiers were trained until the best score among these thresholds would no longer improve, and the model with the maximizing threshold was then used to predict the concepts in the testing set. These procedures were implemented in TensorFlow.

After the classifiers were experimented for all three representations, an attempt to recognize more concepts was made with the AAE representation: a set of new linear classifiers were trained, as above, but for the following 1000 most frequent concepts (after the 500), with even lower thresholds: 0.01, 0.0124, 0.025, 0.0375, 0.05, 0.075, and 0.1. The same procedure was performed again, for the 1000 most frequent concepts following the other 1500. The three prediction files were merged with a concatenation of the concept lists to form this final submission.

2.4.2 Similarity search In this classification method, the images' features were placed in a dense feature index, and the k most similar points were used to determine which concepts are present in each sample. While the use of similarity searching techniques is a naive approach to classification, it enables a rough assessment of whether the representation would fare well in retrieval tasks where similarity metrics were not previously learned, which is the case for the Euclidean distance between features.

Like in logistic regression, the training set was split into two portions 90%/10%, leaving the latter for fine tuning and validation. The Euclidean (L^2) distance was mainly used for determining the similarity between data points. However, as an attempt to exploit the hyperspherical vector space in the features emerging from the AAE and the F-AAE, we have also tested cosine similarity for these representations. For this particular metric, features were linearly normalized to unit norm, so that the internal product could be employed by the index. Faiss [7] was used for the vector similarity search.

A modified form of the k-nearest neighbors (k-NN) algorithm was used: a positive prediction for a given label was made when at least one of the captured neighbors is positive for that label, and false otherwise. This tweak makes the algorithm much more sensitive to very sparse classification labels. The hyperparameter k was exhaustively searched in $1.2, 3, 4, 5$ to yield the highest $F_1 score$ against the validation set. Analogous to the threshold in logistic regression, the optimal k was used to build the predictions on the testing set.

3 Results

The methods described were combined into a total of 9 official submissions from the team, as listed in Table 3. Additional details regarding each run are provided

further below, as separate tables. *Rank* is the final position out of all graded runs from this year's participations in the task. The Kind of representation and classifier type is specified. *Kind* corresponds to the feature extractor used. *Classifier* is either *linear(C)* for logistic regression on the C most frequent concepts, or *k-NN(M)* for similarity search with the given metric. All F_1 scores stated are the micro F_1 scores of each sample averaged across the corresponding set (validation or testing).

Tbl. 3: List of submitted runs from the UA.PT Bioinformatics research group to the concept detection challenge.

Rank	Run file name	Kind	Classifier	**Test** F_1
1	aae-500-o0-2018-04-30_1217	AAE	linear(500)	0.110176
2	aae-2500-merge-2018-04-30_1812	AAE	linear(2500)	0.108229
3	lin-orb-500-o0-2018-04-30_1142	ORB	linear(500)	0.097769
9	faae-500-o0-2018-04-27_1744	F-AAE	linear(500)	0.082475
11	knn-ip-aae-train-2018-04-27_1259	AAE	k-NN(cosine)	0.056958
12	knn-aae-all-2018-04-26-1233	AAE	k-NN(L^2)	0.055936
19	knn-orb-all-2018-04-24_1620	ORB	k-NN(L^2)	0.031376
21	knn-ip-faae-all-2018-04-27_1512	F-AAE	k-NN(cosine)	0.027978
22	knn-faae-all-2018-04-26_0933	F-AAE	k-NN(L^2)	0.027188

Table 4 shows the validation metrics and accompanying final score of the runs based on logistic regression, including which of the tested thresholds yielded the highest score. The *validation* F_1 *score*, which was obtained from evaluating the model against the validation set, only assumes the existence of the respective target concepts (i.e. the 500 most frequent in most runs). Nonetheless, these metrics were assumed to be acceptable for an objective comparison among local runs, and have indeed defined the same ranking order as the final *Test* F_1 *score*.

The adversarial auto-encoder resulted in better quality features than the ORB bags of words or the flipped adversarial auto-encoder. Although the optimal thresholds were low, representations with more descriptive power required less threshold lowering. The second row shows the final outcome of merging the three portions of the multi-label classifiers together (500 + 1000 + 1000). Locally, they were evaluated independently. The extended label set classifiers in this case, albeit covering more biomedical concepts, were less informative on the rarer labels, which ended up crippling the final score.

The work by Lipton et al [23] provide some relevant insights on classifiers aiming to maximize F_1 score. First, that the optimal threshold in a probabilistic classifier which maximizes the score s will be $\max \mathbb{E}_{p(y|s)}[\frac{F_1}{2}]$. Notably, the thresholds identified in our experiments would usually be slightly lower than half the obtained validation F_1 score. The same work presents an algorithm to further

fine-tune these thresholds to attain a more ideal operating point. However, given the higher risks of batch observation (since the thresholds are fitting a portion of the data) and uninformative classifier observation (from the difficuly of classifying rare concepts), we chose to avoid overfitting the validation set by selecting a few thresholds within the interval known to contain the optimal threshold.

Tbl. 4: Results obtained from the submissions to the ImageCLEF 2018 concept detection task with logistic regression.

Kind	Target Concepts	Optimal Thres.	Val. Precision	Val. Recall	Val. F_1	**Test** F_1
AAE	500	0.1	0.216225	0.291748	0.248372	0.110176
AAE	2500	-	-	-	-	0.108229
^	1000	0.05	0.107612	0.100736	0.104060	-
^	1000	0.025	0.079174	0.089273	0.083921	-
ORB	500	0.1	0.213660	0.254010	0.232094	0.097769
F-AAE	500	0.075	0.182600	0.147899	0.163428	0.082475

With the k-nearest neighbors algorithm, five runs were submitted Table 5. It is understandable that logistic regression has a strong vantage point over this method, since it learns a linear vector space that is more ideal for classification, whereas $k - NN$ is restricted to a set of fixed metrics. The resulting scores were significantly lower, but were closed to the final score against the testing set. It is also worth emphasizing that cosine similarity over the features of the AAE and the F-AAE resulted in slightly better metrics.

Tbl. 5: Results obtained from the submissions to the ImageCLEF 2018 concept detection task with similarity search.

Kind	Metric	k	Val. Precision	Val. Recall	Val. F_1	**Test** F_1
AAE	cosine	2	0.065085	0.120610	0.073711	0.056958
AAE	L^2	2	0.062775	0.116644	0.071168	0.055936
ORB	L^2	4	0.023813	0.080327	0.032593	0.031376
F-AAE	cosine	3	0.065085	0.069767	0.031575	0.027978
F-AAE	L^2	3	0.021505	0.062078	0.028007	0.027188

The low performance obtained with the F-AAE in both classification algorithms was empirically justified in Section 2.2.3. However, these results were as low as to suggest that the model would still greatly benefit from better training. From observing generated samples side by side with real images, we have noticed that certain image categories were very underrepresented to non-existent in the

generated samples. Mode collapse is one of the major issues that may arise in GAN training, and is still heavily tackled in recent literature.

4 Conclusion

This paper presents the methods used to obtain unsupervised representations for medical images in literature. We show that the use of deep learning methods can surpass more traditional representations, such as the bags of visual words, in terms of descriptive power. These representations were evaluated by treating the concept detection as a multi-label classification problem, and attained a best mean F_1 score of 0.110176 with logistic regression, ranking first in this year's edition of the concept detection task. A score of 0.056958 was also attained with parameterless vector searching alone. No external sources of evidence were used for any of the presented methods.

On the other hand, these results may not seem to provide a substantial jump when compared to the initial iteration of the ImageCLEF caption challenge. For instance, the best run of the 2017 concept detection task which did not rely on any external resources had a mean F_1 score of 0.1436 [24], and the use of a pre-trained neural network had achieved a score of 0.1583 [25]. Granted, the scores are not directly comparable due to a variety of factors which could influence the performance of concept detection. As a new and disjoint set of medical images and concepts, the quality of the latest data set was slightly improved with the exclusion of multi-figures, and the overall size of the training set was increased by roughly 28%. On the other hand, the number of different concept identifiers presented in the training set's ground truth increased significantly, which may also make the detection task more difficult. The sample-averaged F_1 score for evaluating these solutions is certainly preferable over the macro F_1 score, which would have skewn the scores heavily due to the excessive weight applied to the rare labels [23]. Nevertheless, we find that most concepts in the set do not have enough images with a positive label for a valuable classifier, and that the obtained performance measurements are within the expected range of scores in this task, as the classified labels are very varied and scarce.

Multiple roads to future work can be outlined from this year's participation:

- Generative adversarial networks still make a hot topic, but it is likely to bring remarkable breakthroughs in feature learning. With the emergence of promising techniques for improving the quality and training process of GANs to this purpose, they should likewise be considered for this task and potentially other similar problems.
- There is an open opportunity to learn richer representations with semi-supervised learning, by taking advantage of the concepts availble in the training set. The original paper on the adversarial auto-encoder contemplates one form of incorporating label information in the regularization process [12],

but the currently known approaches to semi-supervised GANs are much more diverse at the time of writing [26, 27].
- We do not exclude the possibility of attaining better scores with other classification algorithms, potentially those which are more adapted to an extreme multi-label classification scenario. The decisions made in this work were based on the desire to test the representation's descriptiveness without learning another complex feature space.

Finally, we find of significant importance that past efforts in the ImageCLEF challenges make a smooth transition to future editions, so as to obtain more competitive baselines and enable new research groups to join in with less technical debt. Delivering the necessary software components to reproduce the results as open source software would contribute to this cause.

Acknowledgements

This work is financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme, and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia within project PTDC/EEI-ESS/6815/2014. Eduardo Pinho is funded by the FCT under the grant PD/BD/105806/2014.

References

1. Ionescu, B., Müller, H., Villegas, M., García Seco de Herrera, A., Eickhoff, C., Andrearczyk, V., Dicente Cid, Y., Liauchuk, V., Kovalev, V., Hasan, S.A., Ling, Y., Farri, O., Liu, J., Lungren, M., Dang-Nguyen, D.-T., Piras, L., Riegler, M., Zhou, L., Lux, M., Gurrin, C.: Overview of ImageCLEF 2018: Challenges, datasets and evaluation. In: Experimental ir meets multilinguality, multimodality, and interaction. LNCS Lecture Notes in Computer Science, Springer, Avignon, France (2018).

2. García Seco de Herrera, A., Eickhoff, C., Andrearczyk, V., Müller, H.: Overview of the ImageCLEF 2018 caption prediction tasks. In: CLEF 2018 working notes. CEUR-WS.org <http://ceur-ws.org>, Avignon, France (2018).

3. Bodenreider, O.: The unified medical language system (UMLS): integrating biomedical terminology. Nucleic acids research. 32, D267—D270 (2004).

4. Pinho, E., Silva, J.F., Silva, J.M., Costa, C.: Towards Representation Learning for Biomedical Concept Detection in Medical Images: UA. PT Bioinformatics in ImageCLEF 2017. In: Working notes of conference and labs of the evaluation forum., Dublin, Ireland (2017).

5. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: IEEE international conference on computer vision (ICCV). pp. 2564–2571. IEEE (2011).

6. Bradski, G., Others: The opencv library. Doctor Dobbs Journal. 25, 120–126 (2000).

7. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. arXiv preprint arXiv:1702.08734. (2017).

8. Li, Z., Zhang, X., Müller, H., Zhang, S.: Large-scale Retrieval for Medical Image Analytics: A Comprehensive Review. Medical Image Analysis. (2017).

9. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision. 60, 91–110 (2004).

10. Goodfellow, I.J., Pouget-abadie, J., Mirza, M., Xu, B., Warde-farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Nets. 1–9 (2014).

11. Radford, A., Metz, L., Chintala, S.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv preprint arXiv:1511.06434. 1–16 (2016).

12. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial Autoencoders. (2015).

13. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer Normalization. (2016).

14. Zhang, J., Dang, H., Lee, H.K., Chang, E.-C.: Flipped-Adversarial AutoEncoders. arXiv preprint arXiv:1802.04504. (2018).

15. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive Growing of GANs for Improved Quality, Stability, and Variation. (2017).

16. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. arXiv preprint arXiv:1605.09782. (2016).

17. Kingma, D.P., Ba, J.L.: Adam: A Method for Stochastic Optimization. In: International conference on learning representations (2015).

18. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: International conference on machine learning. pp. 448–456 (2015).

19. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. arXiv preprint arXiv:1701.04862. (2017).

20. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research. 15, 1929–1958 (2014).

21. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg,

J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. (2016).

22. McMahan, H.B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J., Nie, L., Phillips, T., Davydov, E., Golovin, D., Others: Ad click prediction: a view from the trenches. In: Proceedings of the 19th acm sigkdd international conference on knowledge discovery and data mining. pp. 1222–1230. ACM (2013).

23. Lipton, Z.C., Elkan, C., Narayanaswamy, B.: Thresholding Classifiers to Maximize F1 Score. Machine Learning and Knowledge Discovery in Databases. 8725, 225——239 (2014).

24. Valavanis, L., Stathopoulos, S.: IPL at ImageCLEF 2017 Concept Detection Task. In: Working notes of conference and labs of the evaluation forum. Springer, Dublin, Ireland (2017).

25. Dimitris, K., Ergina, K.: Concept detection on medical images using Deep Residual Learning Network. In: Working notes of conference and labs of the evaluation forum. Springer, Dublin, Ireland (2017).

26. Springenberg, J.T.: Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks. (2015).

27. Kumar, A., Sattigeri, P., Fletcher, T.: Semi-supervised Learning with GANs: Manifold Invariance with Improved Inference. In: Advances in neural information processing systems. pp. 5540–5550 (2017).