Multi-Label and Cross-Modal Based Concept Detection in Biomedical Images by MORGAN_CS at ImageCLEF2020

Oyebisi Layode[1] and Md Mahmudur Rahman[2]

[1] Computer Science Department, Morgan State University, Maryland, USA
oylay1@morgan.edu
[2] Computer Science Department, Morgan State University, Maryland, USA
md.rahman@morgan.edu

Abstract. Automating the detection of concepts from medical images still remains a challenging task, which requires further research and exploration. Since the manual annotation of medical images poses a cumbersome and error prone task, the development of concept detection system would reduce the burdens of annotation, interpretation of medical images while providing a decision support system for medical practitioners. This paper describes the participation of the CS department at Morgan State University, Baltimore, USA (Morgan_CS) in the medical Concept Detection task of the ImageCLEF2020 challenge. The task involves generating appropriate Unified Medical Language System (UMLS) Concept Unique Identifiers (CUIs) for corresponding radiology images. We approached the concept detection task as a multilabel classification problem by training a classifier on several deep features extracted from using pre- trained Convolutional Neural Networks (CNNs) and also by training a deep Autoencoder. We also explored a Recurrent Concept Sequence generator based on using a multimodal technique of combining text and image features for recurrent sequence prediction. Training and evaluation were performed on the dataset (training, validation, and test sets) provided by the CLEF organizer and we achieved our best F1 scores as 0.167 by using DenseNet based deep feature.

Keywords: Medical imaging; Image annotation; Deep learning; Concept detection; Multi-label classification

1 Introduction

Diagnostic analysis of medical images such as radiography or biopsy mostly involve interpretations based on observed visual characteristics. In essence, visual characteristics or features from images can be mapped to its corresponding

semantic annotations. Neural networks over the last two decades have been successfully modeled to learn such mappings from data [1] and consequently this paper involves the annotation of medical images to generate condensed textual descriptions in the form of UMLS (Unified Medical Language System) CUIs (Concept Unique Identifiers) [2] using the dataset under ImageCLEFmed 2020 concept detection task [3], which is a subset of a larger Radiology Objects in COntext (ROCO) dataset [4]. The main objective of this challenge involves automatically identifying the presence of concepts (CUIs) in a large corpus of medical images based on the visual image features. The concept detection task began in 2017 under the ImageCLEF challenge [5] and the participants were tasked with developing methods for predicting captions and detecting the multilabel concepts over a range of medical and non-medical images in a corpus. For example, our previous participation [6] in the ImageCLEFmed 2018 challenge involved the use of LSTM architectures in creating models that approached the concept detection task by developing a language model that predicts the probability of the next word (concept) occurring in a text sequence from the features of an image input and the words (concept) already predicted. This year, the task was limited strictly to concept detection in radiology images [3]. Evaluation criteria for the results obtained is given as the F1 score between the predicted concepts and ground truth concept labels.

1.1 Dataset

The dataset contains 64,753 radiology images from different modality classes as the training set, 15,970 radiology images as the validation and 3,534 radiology images from the same modality classes as the test set [3]. The training images are annotated with 3047 unique UMLS concepts serving as the image captions. The maximum length of the concept annotation is 140 and the minimum annotation is 1. The frequency distribution of the 3047 UMLS concepts across the training images is represented in the Table 1.

Table 1. Concept Frequency in Training set.

Concept Group	Frequency
$>$ 1000	22
500 - 999	298
200 - 499	735
100 - 199	704
60 - 100	704
$<$ 59	785

2 Methods

We approached the concept detection task by comparing elementary CUI multilabel classification and a recurrent CUI sequence generation using extracted features from varying deep learning architectures. The multilabel classification involves feeding the outputs from a feature extraction network into a fully connected network to obtain a sigmoid activation output representing the CUI label predictions.

2.1 Feature Extraction

Feature extraction is a critical component of medical image analysis. The descriptiveness and discriminative power of features extracted from medical images are critical to achieve good classification and retrieval performances. Instead of using any hand-crafted features, transfer learning techniques can be used to extract features of images from a relatively small dataset using pre-trained Convolutional Neural Network (CNN) models [7].

Visual Feature Extraction To perform deep feature extraction, we chose Densenet169 [9] and ResNet50 [8] as our pre-trained CNN models. These models have been trained on the ImageNet [10] dataset consisting of 1000 categories. The Densenet architecture consists of dense blocks of convolution layers - with consecutive operations of batch normalization (BN)[14], followed by a rectified linear unit (ReLU) [15], which provides direct connections from any layer in the block to all subsequent block layers [8]. ResNet, short for Residual Networks is a classic neural network, which is implemented with double - or triple - layer skips that combine features within this residual block of layers and contain nonlinearities (ReLU) and batch normalization in between [8]. We used the Densenet169 and ResNet50 pre-trained models which is a 169 layered dense network and a 50 layered residual network respectively. Both models have been trained on 1.28 million images [8, 9]. For feature extraction, both models are modified to exclude the final 1000-D classification layer and the output before this classification layer is saved. To obtain our deep features, the input images are first reduced to the required input size of 224×224 and further preprocessed using the Keras [16] preprocess_input function, which preprocesses the input into the format the model requires. Since the DenseNet model had been modified to exclude the final 1000-D classification output, a 4096-D feature vector is obtained as the output from the last Average Pooling layer. Also, a 2048-D feature vector was obtained by passing the 224×224 input images through the modified pre-trained ResNet50 model. The extracted features are utilized for transfer learning with multilabel and recurrent CUI sequence classification models built on the Densenet features and a feature fusion of the Densenet and Resnet extracted features.

Feature Fusion Feature fusion methods have been demonstrated to be effective for many computer vision-based applications [11]. Combining features

learned from various architectures creates an expanded feature learning space. We combined the features obtained from the pretrained DenseNet169 and the ResNet50 models by computing the partial least square canonical correlation analysis (PLS-CCA) [17] of both feature vectors, the canonical correlation computes a linear combination of the feature elements from both vectors such that the correlation between the vectors is maximized. Before computing the PLS-CCA, the ResNet50 based deep features are resized from the 2048-D vector to 4096-D output. Since the PLS-CCA required both vectors to be the same dimension the resized 4096-D vector is obtained by doubling each element from the 2048-D vector. The PLS-CCA is computed by combining the 4096-D DenseNet with the resized 4096-D Resnet based deep features. For feature vectors X (4096-D DenseNet) and Y (4096-D Resnet), first and second component vectors u and v are obtained such that the correlation $corr(X, Y)$ is maximized [17]:

$$corr((X, u), (Y, v)) = \frac{u^t \cdot X^t \cdot Y \cdot v}{\sqrt{u^t \cdot X^t \cdot X \cdot u}\sqrt{v^t \cdot Y^t \cdot Y \cdot v}} \tag{1}$$

Where, $u = a_1X_1, a_2X_2 \cdots a_nX_n$ and $v = b_1Y_1, b_2Y_2 \cdots b_nY_n$.

Vectors u and v are obtained by computing the weight vectors $[a_1, a_2 \cdots a_n]$ and $[b_1, b_2 \cdots b_n]$. We selected the first component 4096-D feature vector from the PLS-CCA computation. The result obtained is representative of the features from the maximized correlation of the DenseNet169 and ResNet50 features.

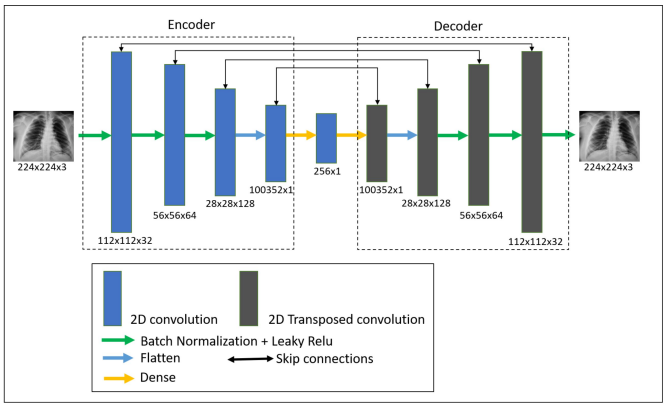

Fig. 1. Encoder-Decoder Architecture

Feature Extraction based on Autoencoder We also use an encoder-decoder-based framework (Fig. 1) to extract deep feature representations unique to the dataset. Autoencoders are a type of unsupervised neural network (i.e., no class labels or labeled data) that consist of an encoder and a decoder model [12]. When trained, the encoder takes input data and learns a latent-space representation

of the data. This latent-space representation is a compressed representation of the data, allowing the model to represent it in far fewer parameters than the original data.

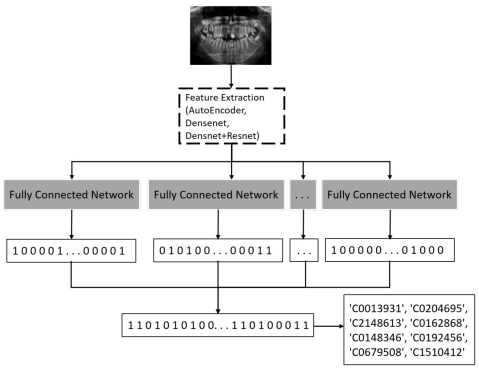

Fig. 2. Multilabel Classification Process Diagram

The encoder region contracts normalized pixel-wise data from input images into smaller dimensional feature maps using sequential layers of 2D convolutions, batch normalization and ReLU activation. The output from the convolutional blocks is passed to a fully connected layer that represents a 256-D feature space. The decoder expands the 256 fully connected output by applying transposed convolutions that up sample the features back to the original input size. Batch normalization and ReLU activation are also added at each step of the transposed convolution sequence and the encoder filter sizes mirror the decoder filter sizes. The 256-feature output from the encoder is given as the auto-encoded deep feature representation of our input image. The Autoencoder was trained using the Adam optimizer [18] and a mean squared error loss on the ROCO training dataset for 20 epochs with a batch size of 50. The initial Adam learning was also set to 0.001.

Text Feature Extraction The deep text features are extracted from the image concepts by learning and mapping deep feature embeddings that represent the sequence of image concepts. The embeddings are learned during training when a fixed length of CUI sequence is passed to a neural embedding layer. Before passing the CUI sequences to the embedding layer, for each input image, the image concept sequence is tokenized using the Keras text preprocessing library. Since a fixed length of tokenized CUI sequence was required for the embedding layer the differences in CUI sequence length for different input images was accommodated by zero-padding the tokenized sequence up to the maximum CUI sequence length of 140. During training, the embedding layer uses a mask to ignore the padded values and its output is passed to a long short-term memory

(LSTM) layer [13] with 256 memory units. The output from the text encoding block of the embedding and LSTM layer is a 256-D vector holding recurrent information that may be mapped back to the input concept sequence.

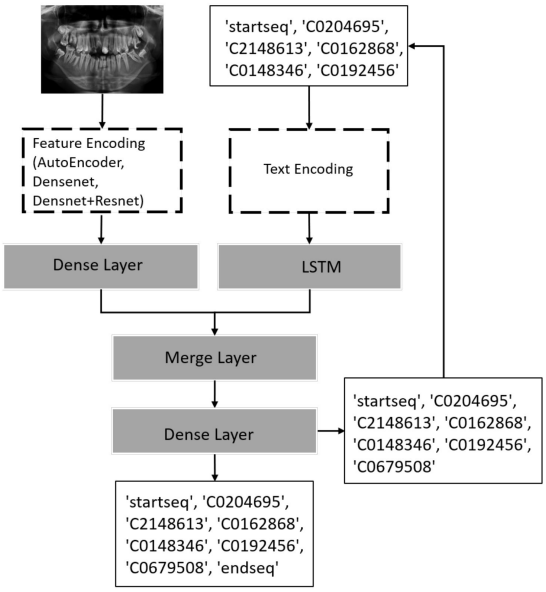

Fig. 3. Recurrent CUI Sequence Generator

2.2 Multi-label Classification

The high volume of classification (CUI) labels (3047) and imbalance in the label frequency results in a huge bias towards the multi-label classification problem. The concepts set was split into groups based on the concept frequencies (Table 1) and separate models were trained for classification within the concept set groups. The DenseNet feature, fused DenseNet-ResNet feature and the Auto-encoded feature are passed to a stack of fully connected layers for the multi-label prediction in the different dataset groups as shown in Fig. 2. The fully connected network is composed of Dense layers stacked together to learn weights for a final sigmoid classification of the concept labels. The expected input for the fully connected classifier is the deep encoded feature vector corresponding to an image while the output is the binary multi-label classification of the concepts associated with the input image features.

The fully connected classifier was trained over 20 epochs with a learning rate of $1e-3$ for an Adam optimizer. Since the concept set was split into groups and a different classifier trained for each concept group the overall CUI prediction

for an input image involves the combination of the predictions from all concept group classifiers.

2.3 Concept Sequence Generation

The CUI sequence generator involves training a recurrent classifier on a fusion of the extracted image features and the embedded textual features. The text features are obtained by learning the embeddings at training time from the embedding layer stacked with a LSTM layer to give a 256-D text feature output. Since concatenating the image and text feature vectors would require equal feature vector lengths, to combine the 256-D text features and the 4096-D image features, the 4096-D image feature is down-sampled to 256 by passing it through a dense layer with 256 units to give a 256-D feature output. The 256-D image feature and 256-D text feature is passed to a concatenation layer to obtain a 256-D output that is passed to a final dense classification layer for the prediction of the next word in the CUI sequence. The CUI sequence prediction begins when a start signal is passed as the first element in the CUI sequence and the prediction ends when a stop signal is predicted by the classification model as shown in Fig. 3. The recurrent classifier was trained over 30 epochs with a learning rate of $1e-3$ for an Adam optimizer and a batch size of 50.

3 Results and Discussions

Using the provided test dataset, multiple runs were submitted based on the multi-label classification with DenseNet, DenseNet-ResNet and Auto Encoded features. The result from the recurrent concept sequence generator with DenseNet encoded features was also submitted and the F1 evaluations are represented in Table 2. Our best result with a F1 score of 0.167 was obtained from the multi-label classification of DenseNet feature.

Table 2. F1 scores of submitted run (test set).

Run	Method	F1 Score
MSU_dense_fcn	Densenet169 + multilabel classification	0.167
MSU_dense_resnet_fcn_1	(Densenet169 + Resnet50) + multilabel classification	0.153
MSU_dense_feat	Densenet169 + multilabel classification	0.139
MSU_dense_fcn_2	Densenet169 + multilabel classification	0.094
MSU_dense_fcn_3	Densenet169 + multilabel classification	0.089
MSU_autoenc_fcn	Autoencoder + multilabel classification	0.063
MSU_lstm_dense_fcn	Desnet169 + Recurrent concept generator	0.062

1. **MSU_dense_fcn:** This run utilized a multi-label classification model with the training parameters (described in section 2.2) based on the features extracted from a pre-trained DenseNet169. The threshold for the prediction

score is set at 0.4 for the multi-label sigmoid classification which ranged from 0 to 1. Concept labels with prediction scores less than 0.4 are considered irrelevant to the input image.
2. **MSU_dense_resnet_fcn_1:** In this run, the PLS-CCA of DenseNet169 and ResNet50 features ar computed to obtain fused features for the multi-label classification. The prediction score threshold for this run is also set at 0.4 for the final multi-label sigmoid classification
3. **MSU_dense_feat, MSU_dense_fcn_2, MSU_dense_fcn_3:** These runs are variations of the **MSU_dense_fcn** run with different prediction score thresholds of 0.5, 0.3 and 0.25 respectively.
4. **MSU_autoenc_fcn:** The encoder-decoder model is utilized for this run to obtain the encoded features of the input images. The multi-label classification model (with parameters same as in runs 1,2 and 3) is trained on the Autoencoded features. The threshold for the prediction score from the classification model is also set to 0.4.
5. **MSU_lstm_dense_fcn:** This run involved the recurrent generation of concepts by utilizing image features extracted from DenseNet169 combined with embedded concept sequences as described in 2.3. The obtained results clearly show the concept prediction challenge as more of a classification problem than a sequence generation task since all multi-label classification approaches performed better.

4 Conclusions

This article describes the strategies of the participation of the Morgan CS group for the concept detection tasks of ImageCLEF2020. We performed multi-label classification of CUIs in different deep feature spaces. We achieved comparable results considering the limited resources (computing and memory power) we had at the time of the submission. Since the ROCO data set is grouped into different modalities, we plan to perform separate multi-label classification for the different modalities in future.

Acknowledgment

This work is supported by an NSF grant (Award ID 1601044), HBCU-UP Research Initiation Award (RIA).

References

1. O. Vinyals, A. Toshev, S. Bengio and D. Erhan : Show and tell: A neural image caption generator. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, pp. 3156-3164, (2015) https://doi.org/DOI: 10.1109/CVPR.2015.7298935
2. O. Bodenreider : The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Res. **32**(5), D267–D270 (2004)

3. O. Pelka, C. M. Friedrich, A. García Seco de Herrera and H. Müller: Overview of the ImageCLEFmed 2020 Concept Prediction Task: Medical Image Understanding. CEUR Workshop Proceedings (CEUR- WS.org), ISSN (2020)
4. O. Pelka, S. Koitka, J. Rückert, F. Nensa und C. M. Friedrich : Radiology Objects in COntext (ROCO): A Multimodal Image Dataset. Proceedings of the MICCAI Workshop on Large-scale Annotation of Biomedical data and Expert Label Synthesis (MICCAI LABELS 2018), Granada, Spain, September 16, 2018, Lecture Notes in Computer Science (LNCS) **11043**, pp 180–189, (2018) https://doi.org/doi: 10.1007/978-3-030-01364-6_20
5. B. Ionescu, H. Müller, R. Péteri, A. Ben Abacha, V. Datla, S. A. Hasan, D. Demner-Fushman, S. Kozlovski, V. Liauchuk, Y. Dicente Cid, V. Kovalev, O. Pelka, C. M. Friedrich, A. García Seco de Herrera, V. Ninh, T. Le, L. Zhou, L. Piras, M. Riegler, P. Halvorsen, M. Tran, M. Lux, C. Gurrin, D. Dang-Nguyen, J. Chamberlain, A. Clark, A. Campello, D. Fichou, R. Berari, P. Brie, M. Dogariu, L. Daniel Ștefan, M. Gabriel Constantin : Overview of the ImageCLEF 2020: Multimedia Retrieval in Lifelogging, Medical, Nature, and Internet Applications In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the 11th International Conference of the CLEF Association (CLEF 2020), Thessaloniki, Greece, LNCS Lecture Notes in Computer Science, 12260, Springer, September 22-25, (2020).
6. M. Rahman : A cross modal deep learning based approach for caption prediction and concept detection by CS Morgan State. Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018, CEUR Workshop Proceedings, **2125**, CEUR-WS.org, (2018)
7. K. Simonyan, A. Zisserman: Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR, **abs 1409.1556** (2014)
8. K. He, X. Zhang, S. Ren and J. Sun :Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 770–778, (2016) https://doi.org/doi: 10.1109/CVPR.2016.90.
9. G. Huang, Z. Liu, L. Van Der Maaten and K.Q. Weinberger :Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, pp. 2261–2269, (2017) https://doi.org/doi: 10.1109/CVPR.2017.243.
10. J. Deng, W. Dong, R. Socher, L. Li, K. Li and L. Fei-Fei:ImageNet: A Large-Scale Hierarchical Image Database. 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, pp. 248-255, (2009). https://doi.org/10.1109/CVPR.2009.5206848
11. T. Akilan, Q. M. J. Wu, Y. Yang and A. Safaei :Fusion of transfer learning features and its application in image classification. 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Windsor, ON, pp. 1–5, (2017) https://doi.org/doi:10.1109/CCECE.2017.7946733.
12. P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio and P. Manzagol: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. The Journal of Machine Learning Research. **11**, pp. 3371–3408 (2010)
13. S. Hochreiter and J. Schmidhuber : Long short-term memory. Neural Computation. **9(8)**, pp. 1735–80 (1997), https://doi.org/DOI: 10.1162/neco.1997.9.8.1735
14. S. Ioffe and C. Szegedy: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ICML'15: Proceedings of the 32nd International Conference on International Conference on Machine Learning. **37**, pp. 448—456, (2015)

15. R.H.R. Hahnloser, R. Sarpeshkar, M.A. Mahowald, R.J. Douglas and H.S. Seung: Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature. **405**, pp. 947–951 (2000). https://doi.org/doi: https://doi.org/10.1038/35016072
16. F. Chollet: keras, GitHub. `https://github.com/fchollet/keras`. Last accessed 29 Jul 2020
17. H. Hotelling : Relations Between Two Sets of Variates. in Breakthroughs in Statistics: Methodology and Distribution, S. Kotz and N. L. Johnson, Eds. New York, NY: Springer, pp. 162—190 (1992)
18. D.P. Kingma and L.J. Ba : Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG] International Conference on Learning Representations (ICLR), (2015)