Techniques for Medical Concept Detection from Multi-Modal Images

Rohit Sonker, Ayush Mishra, Palvika Bansal and Anup Pattnaik

PricewaterhouseCoopers US Advisory, Mumbai, India
{rohit.sonker,ayush.mishra,palvika.lnu,anup.a.pattnaik}@pwc.com

Abstract. With the increasing availability of medical images coming from different modalities (X-Ray, CT, PET, MRI, Ultrasound, etc.), the task of automatic medical image captioning is emerging as a key component in medical research. ImageCLEF 2020 is dedicated to extracting relevant concepts from a large corpus of radiology medical images with different image modalities by learning the visual contents of the images. The variability between modalities and expertise required in interpreting radiology images often represents a bottleneck in clinical diagnosis pipelines. Therefore, we propose a reliable automatic classification method which is highly desired as assistance for human radiologists in producing reports more accurately and efficiently. Throughout the experiment, we leveraged CNN Architectures, NLP, and clustering techniques to come up with our best system. In this paper, we introduce a novel technique of band classification, where we first cluster the vocabulary of concepts into bands and then build customized classification architectures for each of the band. Predictions of one band are given as input to subsequent bands to aid the learning of associated concepts. Also, we systematically explored several pre-processing approaches to handle variations in contrasts, intensities across images of different modalities. In the final evaluation of ImageCLEF 2020, we submitted 9 runs out of which our best systems ranked 3rd, 4th, 5th. Overall our team ranked 2nd among 41 participants globally.

Keywords: Image Captioning · Medical Imaging Modalities · Deep Learning · Machine Learning · Concept Detection · Information Retrieval

1 Introduction

The healthcare industry has been witnessing an increasing shift towards digitization across the world. With more and more hospitals now saving their patient data and medical images electronically, the platform is set to leverage AI capabilities to assist doctors in their diagnosis and enhance the entire healthcare

ecosystem. Medical Images, ranging from MRI scans, CT scans, PET scans, X-Ray are used for diagnosis and treatment of many diseases such as cancer, pneumonia, and pneumothorax. Medical domain experts go through the medical scans of the patient and subsequently write a condensed textual report, which is a time-consuming process and also leads to an increase in the cost of such treatment. The reading and interpretation of medical images, like all other human processes, are prone to error.

The above stated problems and the abundance of medical images in the current scenario have motivated us to use AI techniques to semi-automate the report generation process. We intend to build a pipeline that will take a medical image and caption out the keywords stating the abnormalities and the type of machine used to produce the image. The output caption will not be a free flowing text in Natural Language which would make sense but just a list of keywords next to each other. We hope this system will be an efficient secondary check for the medical experts and will also speed up the report writing process.

ImageCLEF [4] hosted the 4th edition of its Medical Image Captioning task where we were provided with a subset of Radiology Objects in Context (ROCO) dataset [8]. We have used multiple approaches to tackle the problem leveraging deep learning architectures and NLP techniques. Before passing the image into the deep neural network models, they are first passed through certain preprocessing steps, which are further explained in section 3.

We have used the following methods to extract the keywords from a given patient medical image -

1. ResNet18 fine-tuned + Custom CNN
2. ResNet18 on Scan Type
3. Band Classification
4. KNN with ResNet101 embeddings Modality wise
5. KNN on ResNet18 embeddings with weighted label combination
6. Concept Clustering based on data segregation

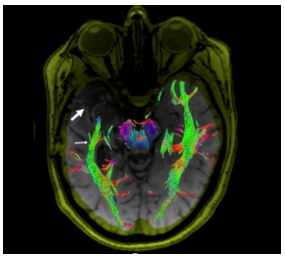
We will explain all the above mentioned methods in detail in Section 4.

2 Data

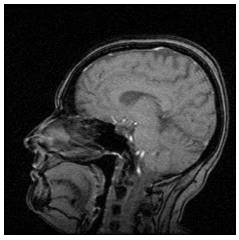
ImageCLEF 2020 MedCaption task [7] focused on extracting information from radiology images. The dataset provided as part of the challenge is a subset of the extended ROCO dataset, with additional imaging modality information. A total of 6,031,814 image-caption pairs were extracted. To focus on radiology images and non-compound figures, automatic filtering with deep learning systems as well as manual revisions were applied. Post the filtering, a total of 80,183 images were provided to us, out of which 64,753 images were part of the training set and 15,970 images were part of the validation set. There were 3047 unique concepts in the training set. For a given image, the number of concepts were in the range of 1-140. The average number of concepts per image was close to 6. All the

images were in jpeg format. The number of channels within the images was not consistent.

There were several challenges within the dataset. There was significant noise in most of the images. The noises ranged from doctor signatures, patient ID and other random numbers inscribed on top of the radiology images. While most of the images were restricted to a given organ of the body (fig.1), some of the images had the entire human body and a zoomed-in scan of a particular organ. Extracting captions from these types of images would be quite difficult as the AI system has to focus only on the zoomed-in part to understand the abnormalities. The zoomed scan in such images occupied very limited space making it more difficult to focus on the relevant part. Many images also had arrows, straight lines, and the watermark of the hospital or organization where the scan was taken, as shown in fig.2.

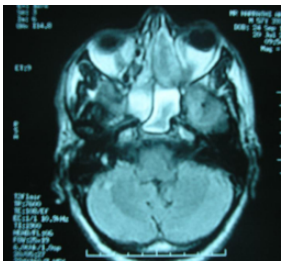


(a) Image: ROCO2_CLEF_05912

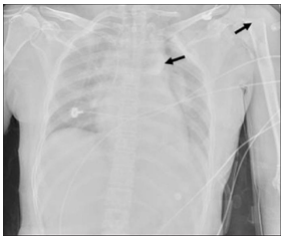


(b) Image: ROCO2_CLEF_31469

Fig. 1: Images from ImageCLEFmed Caption 2020



(a) Image: ROCO2_CLEF_32751



(b) Image: ROCO2_CLEF_58524

Fig. 2: Images from ImageCLEFmed Caption 2020

The data was quite skewed in terms of the number of images in which a particular concept occurred. The range varied from 34 to 20031 which depicts the level of skewness. As part of exploration, we also extracted the text descriptions

of the concept IDs that were provided to us. The caption was processed using QuickUMLS [9] to produce the gold UMLS concept unique identifiers (CUIs). We have further used the text extracted in one of the techniques. However, the text description for 12 concepts were not available.

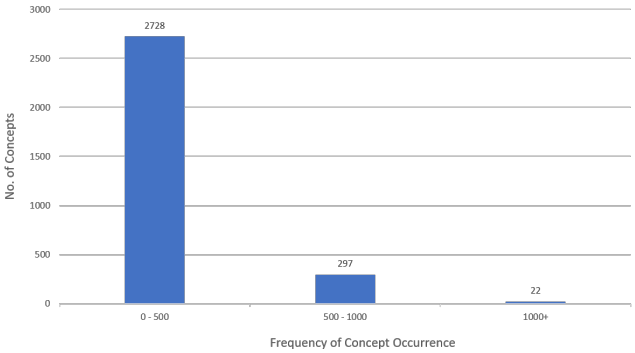


Fig. 3: Frequency of concept occurrence in training images

Fig.3 shows the distribution of the concepts. There are very few concepts with frequency>1000 in the training set and most of them are in the 0-500 bucket. Concepts follow a similar distribution in the validation set as well.

Table 1 shows list of the top 10 most frequently occurring concept IDs along with their descriptions:

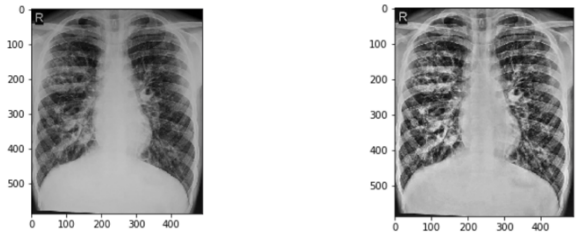
Table 1: Top 10 concept in training set by frequency

Concept ID	UMLS Term Description	No. of Images
C0040398	Tomography, Emission-Computed	20031
C0040405	X-Ray Computed Tomography	20031
C0043299	Diagnostic Radiologic Examination	18944
C0024485	Magnetic Resonance Imaging	11447
C0041618	Ultrasonography	8629
C0002978	Angiogram	4713
C0018792	Heart Atrium	1262
C0021853	Intestines	1219
C0025066	Mediastinum	1205
C0227665	Both kidneys	1186

3 Data Pre-processing

We explored different pre-processing approaches for different systems of models. In general the CLAHE technique was used in most of the systems.

Contrast Limited Adaptive Histogram Equalization (CLAHE) : Images of different modalities have varying brightness, intensity, contrasts, etc, To handle these modalities, and to enhance feature detection, we used the CLAHE [11] technique as the first step. CLAHE equalizes brightness and contrast among images. An image is divided into regions and each region is histogram equalized. To limit noise amplification, contrast limiting is applied. It strengthens feature extraction from the edges of each region in the image. In this, each pixel is transformed based on the histogram surrounding the pixel. CLAHE limits the amplification by clipping the histogram at a predefined value called a clip limit. During our experiments, we used the clip limit of 2.5, as we got the best results from this value. Example is shown in fig.4

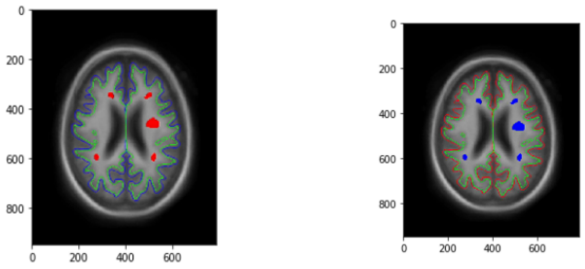


(a) Before (b) After

Fig. 4: CLAHE processing on Image ROCO2_CLEF_57065

Intensity Normalization : MRI images have larger intensity variations, due to different scanners or parameters used during MRI image acquisition. To normalize the intensity, Linear normalization was applied to MRI images. This changes the range of pixel intensity values. We transformed images to new intensity values in the range (0, 255). The motivation behind this was to bring all the images in a similar intensity range that is normal. This helps the deep learning network to learn faster compared to CLAHE. An example is shown in fig.5

Range Normalization : We leveraged simple range normalization for the CT images, where intensities are comparable across different scanners and feature extraction from these images benefits from clipping or rescaling. This normalization mapped intensities to [-1,1] using below mentioned transformation, where shift and scale were identified from minimum and maximum intensities. The



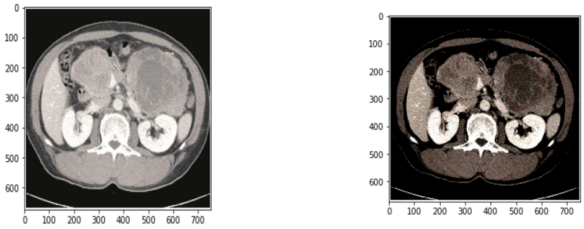
(a) Before (b) After

Fig. 5: Intensity Normalisation on Image ROCO2_CLEF_31504

images were reconstructed through this transformation to handle contrasts. For intensity I,

$$I_{norm} = (\frac{I - shift}{scale}) \times 2 - I \tag{1}$$

The effect of range normalization is shown in fig.6.



(a) Before (b) After

Fig. 6: Range Normalisation on Image ROCO2_CLEF_06408

Data Augmentation : To increase the diversity of data available for training the models, various data augmentation techniques were used in the process. The image is first rotated in the range of 20 degrees after which the image is translated horizontally by 0.2 fractions of total width. Then shear transformation is applied to the image in the counter-clockwise direction by 0.2 radian angle. Then the image is zoomed to a scale ranging from 0.8 to 1.2 and then the image is flipped horizontally. All of the above mentioned transformations are done using Keras ImageDataGenerator on a random basis.

4 Techniques

The input images were first all transformed into 3 channel RGB images and then resized to $224 \times 224 \times 3$ which is the input requirement shape for ResNet architectures [3]. Then the images were passed through a series of pre-processing steps for noise removal and enhancement of brightness and contrast. We have used Keras ImageDataGenerator in all of our techniques to load the images on the go and avoiding the need to load all the training images at once, which will require a lot of disk space. We also performed data augmentation using Keras DataGenerator. The images in the training as well as the validation set were given in 7 different folders, each representing the scan type of the images. Since each image could have multiple CUIs tagged to it, we create a CUI index dictionary mapping concept IDs to integers ranging from 0 to 3046. We created target vectors for each image as a 3047 long vector. The vector will have 1 when the concept ID corresponding to the index of the vector is present in the caption of the image. All the operations were done using Python 3.6.2 and Keras framework. The Deep Learning models were trained on a virtual Ubuntu server machine equipped with 2 NVIDIA Tesla P4 GPU accelerators. The GPUs were accessed using the Google Cloud Platform. We now describe each technique variant in detail below.

4.1 System 1: ResNet18 on All Data

Training Images varied a lot from each other across scan types as well as within the same type of scan. Despite using multiple pre-processing steps to remove noise, performing a multilabel classification with 3047 images was a daunting task. A convolutional neural network with enough depth was the obvious choice to learn the features and patterns and perform the above mentioned task. We chose a pre-trained ResNet18 as our baseline model. In this particular approach, we keep all the layers trainable. We start-off with weights trained on ImageNet data [2]. These weights are trained on images that are completely different from medical images, but they are still a better place to start with as compared to random weights, assuming that the model will be able to extract the high-level features of the image and learn the custom features when trained on our medical images. We took the output of the penultimate layer of ResNet18 and then passed that tensor through a convolutional layer and maxpool layer and finally through 3 fully connected layers. We used sigmoid function as the activation for the last layer to enable multi label outputs. We used binary cross entropy as the loss function and Adam optimizer [5] with a learning rate of 0.0001. The batch size was fixed at 32. The model was trained for 50 epochs and we observed the validation F1 score saturated post 42 epochs. Training each epoch took close to 840 seconds. The model architecture is described in fig.7.

4.2 System 2: ResNet18 on Scan Type

The given dataset had seven types of scanned medical images in their respective folders. A type of scanned image might have different features than another one

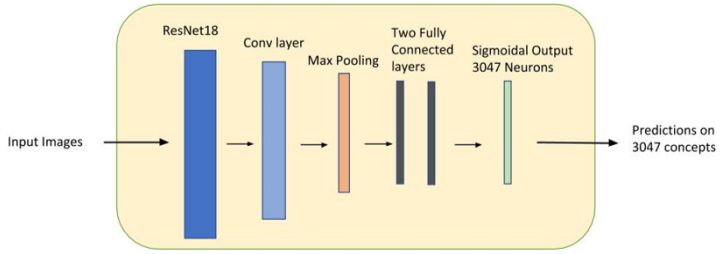


Fig. 7: Model Architecture for ResNet18 trained on entire data

due to different scanners or the way the image is captured, for example, MRI image will have larger intensity variations compared to CT image. Our approach in this method was to train individual folders (based on the type of scan) on a ResNet18 network so that the model better captures the input features from the given image and those features will be limited to the type of scan. The number of images per folder varied from 500 in PET scans to 20000 in Ultrasound scans. The two folders with less than 1000 images (DRCO and DRPE) were merged together and passed through a single ResNet18 network, others were passed into their respective network. The custom layers of all the model had the following custom layers: Conv2d(128,3,3) + MaxPool(2,2) + FC(1024) + FC (512) + FC(No. Of Concepts) + Sigmoid
The model architecture is described in fig.8. The threshold value to calculate the label presence is calculated individually for each network by checking the best F1 score on the validation set.

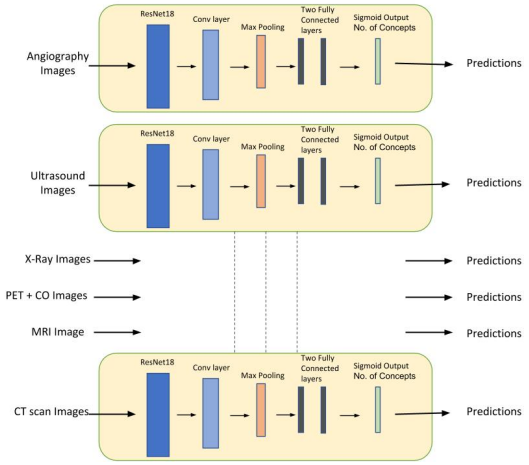


Fig. 8: Model Architecture for ResNet18 trained on Scan Type

4.3 System 3: Band Classification

In this approach, we aim to separately handle the concepts which are predicted correctly versus the concepts which are not being predicted well. The idea behind this approach is to use different network parameters for different sets of labels and to allow correlated labels to enhance performance. The bands consist of different sets of labels categorized by the prediction performance on the complete ResNet18 network. The images are preprocessed using CLAHE and the training set is augmented.

Hence, we first filter out the concepts which are being predicted well by our overall ResNet18 model. We first calculate the F1 score corresponding to each concept and filter out concepts that have an F1 score > 0.2. This forms our first band of target concepts. Subsequently, the remaining concepts are considered as band 2. Based on our threshold of 0.2, band 1 contains 25 concepts and band 2 contains 3022 concepts. Further decomposition to more bands is also possible by repeating the process however, we notice that in band 2, there is no wide separation as to which concepts perform well, hence, no further decomposition is considered.

We train two separate neural networks to predict the concepts in band 1 and band 2, for all images. The images are first passed through band 1 network and their predictions are generated. In the second band, the predictions of the first band are added as an auxiliary input to the network. This is done to include associative information between concepts which may aid in learning concepts that were not being predicted well in a combined network. We use a ResNet 18 model with additional layers for both the band networks. The networks vary slightly in their architecture, due to the addition of more layers and an auxiliary input in band 2.

In band 1, the ResNet18 architecture (without top layer) is followed by a 2D Convolution layer (128,3,3), max pooling (2,2), and two fully connected layers of 512 and 256 neurons respectively. The fully connected layers have ReLU activation. All additional layers have a dropout of 0.2 while training. Finally, we have an output layer (25 neurons) with sigmoidal activation.

In band 2, we have a similar structure with ResNet18 (without top layer) followed by a 2D Convolution layer (128,3,3) with max pooling (2,2) and fully connected (FC) layers of 1024 neurons each. We also have an auxiliary input which is the prediction result of band 1. The input is passed through a layer of 256 neurons and concatenated to the FC layer as mentioned above. Finally, the combined input is passed through an FC layer with 1024 neurons and attached to a sigmoid output layer of 3022 neurons. The architecture is shown in fig.9.

The outputs from both are probabilities of each label. To convert this to a one-hot encoded vector we set a threshold based on the maximum F1 score on the validation set. The outputs from both bands are combined to get the complete prediction for 3047 concepts. This process happens in both bands. The value of the threshold used in band 1 and band 2 was 0.3 and 0.25 respectively.

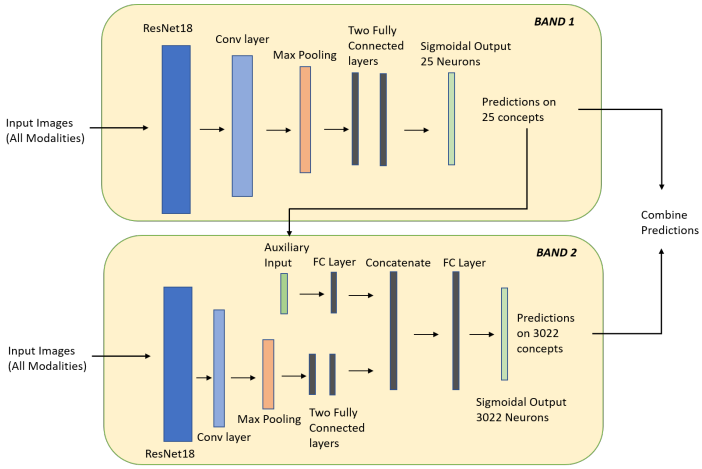


Fig. 9: Model architecture in band classification model

4.4 System 4: K-Nearest Neighbour with ResNet101 Embeddings Modality wise

In this approach, we use K-Nearest Neighbour algorithm [1] on ResNet101 embeddings [3]. For each test image, the K-most similar images from the training set are retrieved and their labels are used to predict labels of the test image. This approach is implemented independently for each modality.

The training images are first converted to embeddings using a ResNet101 encoder. We use ResNet101 pre-trained on ImageNet data, without further training of the network. The input images are preprocessed using CLAHE. No augmentation of the training set is performed here. The embeddings of dimension (7,7,512) are extracted from ResNet101 and then flattened.

These training embeddings are added as a new layer to the encoder network, such that for each input image, we get a similarity value corresponding to each training image. This architecture allows us to compute the cosine similarity of each input with respect to training images. The architecture is depicted in fig.10. Hence, each test image is first encoded and then its cosine similarity with respect to each training image embedding is computed by the network.

To compute the output labels the following approach is used. The labels of these K images are taken as one-hot encoded vectors and first added then normalized. Hence, get a vector with values ranging from 0-1 for all labels. We then select a threshold value t to convert this combined vector to a one-hot encoded label vector. Note, that this process is done separately for each modality. Hence, we have a total of 7 models each with their own values of K and t. K is chosen in proportion to the size of the training set. The threshold value is chosen by analyzing the best performance on the validation set. The details of K,t chosen are in the table 2.

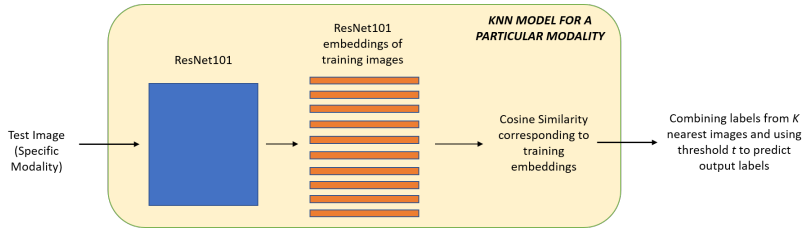


Fig. 10: Model architecture for KNN with ResNet101 embeddings

Table 2: Number of neighbours K and threshold value t for each modality

Modality	No. of Neighbours K	Threshold t
DRMR	200	0.20
DRXR	200	0.26
DRAN	50	0.20
DRUS	100	0.22
DRCT	200	0.20
DRCO	25	0.35
DRPE	25	0.29

4.5 System 5: KNN on ResNet18 Embeddings with weighted label combination

In this approach, we use a K-Nearest Neighbour algorithm on ResNet18 embeddings. A single ResNet18 network is used here for all modalities. The images are preprocessed using CLAHE and the training set is augmented using the methodology described earlier. The ResNet18 network is first trained on the training set as described in the section on ResNet18 (all data) and then the top layer is removed. This forms the encoder network.

The training images are converted into embeddings using the encoder and these embeddings are used as layers on top of the encoder network. This leads to architecture similar to the previous approach however all modalities are taken together here. A test image is first encoded and its cosine similarity to all training images is computed. We select a K of 200 and all label vectors of these closest training images are retrieved.

In this approach, we define a new method to combine these K one-hot label vectors. We use a method similar to term frequency-inverse document frequency (TF-IDF) to combine values for each label. The combined value is proportional to the occurrence of a label in the K images and inversely proportional to their occurrence in the complete training set. This allows rare concepts to be predicted more. The formula is described below. Let L be the combined vector of the sum of K one hot encoded label vectors of the nearest neighbours for a particular test point. Let R be the resulting combined label vector for our test point. The

for each label i the value R_i is defined as -

$$R_i = \frac{L_i}{max(L)} \times log(\frac{size\ of\ training\ set}{frequency\ of\ label\ i\ in\ training\ set}) \tag{2}$$

Once the vector R has been computed, we must convert this vector of values to a one-hot encoded vector to predict labels. For this, we set a threshold value which gives the best performance on the validation set, hence threshold $t = 1.17$.

4.6 System 6: Concept Clustering based data segregation

The Concept Unique identifier (alphanumeric code for concepts) associated with the image must have some relation with each other, for example, an aneurism would be more closely related with blood clot than a fractured bone. In this approach, we tried to group together such concepts that were closely related to each other. However, this relationship would be hard to determine using CUIs which were assigned to a given image, hence we converted the unique identifiers into human readable comments using UMLS conversion [9]. Once we had the UMLS converted concepts, we extracted embeddings for each using BioWordVec [10] which is a word to vector converter trained on vocabulary frequently used in the medical field. We then calculated the closeness of a concept with respect to others using cosine similarity and then grouped them into 6 clusters using the k-means algorithm [6].

Once we have a data table with similarity scores of concept with each other, it is divided into 5 clusters using k-means clustering. Each cluster had between 30,000 to 42,000 images and 400 to 600 concepts associated with it, except for the 6th cluster which had all the images with concepts whose embedding was not found using BioWordVec which had approximately 26000 images and 170 concepts. The process is described in fig.11.

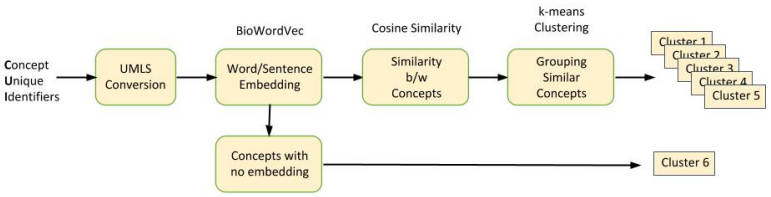


Fig. 11: Process to generate concept clusters

A data frame was created for each cluster and was trained on ResNet18 model (w/o top layer) and following custom layers: Conv2d(128,3,3) + MaxPool(2,2) + FC(1024) + FC (512) + FC(No. Of Concepts) + Sigmoid

All the clusters were trained individually with a dedicated model. The architecture is shown in fig.12. The test image was passed through each of these

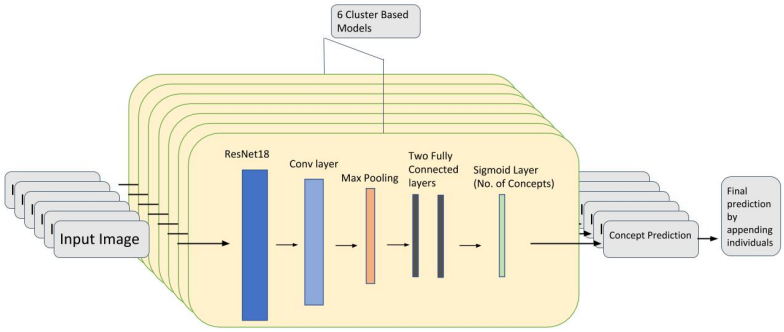


Fig. 12: Model Architecture for Concept Clustering Approach

6 models and their predictions were appended to make the final prediction of concepts.

5 Results

We submitted our concept predictions on the test set in a txt file, in which each row corresponds to the image ID followed by the predicted Concept IDs of that image via our models. The predictions were evaluated using F1 score, by comparing the ground truth vector y_{true} to the predicted concept vector y_{pred} and then averaging across all test images. Both the vectors were 3047 in length, which is equal to the number of unique classes present. The KNN clustering using ResNet-101 embeddings gave the best F1 score of 0.392. All results are shown in table 3.

Table 3: Results of different methods on test data

System	Technique	F1 Score
1	ResNet18 across all data	0.368
2	ResNet18 on Scan Type	0.389
3	Band Classification	0.338
4	**KNN on ResNet101 Embeddings Modality wise**	**0.392**
5	KNN on ResNet18 Embeddings with Weighted label combination	0.366
6	Concept Clustering based data segregation	0.316

6 Conclusions and Future Work

Throughout the challenge, we experimented with visual features of images as well as the UMLS embeddings and tried to benefit from the concept grouping through the clustering mechanism. The setup is motivated by the variabilities in modalities of radiology images and stimulus to extract value from all the data that is provided. Our best model KNN with ResNet101 embeddings achieved an F1 score of 0.39238 and ranked 3rd. The band classification approach that we introduced in this paper allows the usage of different network architectures for different sets of labels. Incorporating predictions from a set of bands in making predictions for other bands promises an increase in performance in scenarios where the concepts in bands are associated with each other.

In future work, we aim to experiment with the attention mechanism to focus on important features of images to improve concept detection and interpretability of the models. We also aim to improve the performance of the band classification approach. Exploring skewness in data availability for different concepts could be an interesting extension to our models.

References

1. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory **13**(1), 21–27 (1967)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
4. Ionescu, B., Müller, H., Péteri, R., Abacha, A.B., Datla, V., Hasan, S.A., Demner-Fushman, D., Kozlovski, S., Liauchuk, V., Cid, Y.D., Kovalev, V., Pelka, O., Friedrich, C.M., de Herrera, A.G.S., Ninh, V.T., Le, T.K., Zhou, L., Piras, L., Riegler, M., l Halvorsen, P., Tran, M.T., Lux, M., Gurrin, C., Dang-Nguyen, D.T., Chamberlain, J., Clark, A., Campello, A., Fichou, D., Berari, R., Brie, P., Dogariu, M., Ştefan, L.D., Constantin, M.G.: 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), vol. 12260. LNCS Lecture Notes in Computer Science, Springer, Thessaloniki, Greece (September 22-25 2020)
5. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
6. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. vol. 1, pp. 281–297. Oakland, CA, USA (1967)
7. Pelka, O., Friedrich, C.M., García Seco de Herrera, A., Müller, H.: Overview of the ImageCLEFmed 2020 concept prediction task: Medical image understanding. In: CLEF2020 Working Notes. CEUR Workshop Proceedings, CEUR-WS.org, Thessaloniki, Greece (September 22-25 2020)

8. Pelka, O., Koitka, S., Rückert, J., Nensa, F., Friedrich, C.M.: Radiology objects in context (roco): A multimodal image dataset. In: Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis, pp. 180–189. Springer (2018)
9. Soldaini, L.: Quickumls: a fast, unsupervised approach for medical concept extraction (2016)
10. Yijia, Z., Chen, Q., Yang, Z., Lin, H., lu, Z.: Biowordvec, improving biomedical word embeddings with subword information and mesh. Scientific Data **6** (12 2019). https://doi.org/10.1038/s41597-019-0055-0
11. Zuiderveld, K.: Contrast limited adaptive histogram equalization. Graphics gems pp. 474–485 (1994)