Concept Detection and Caption Prediction in ImageCLEFmedical Caption 2023 with Convolutional Neural Networks, Vision and Text-to-Text Transfer Transformers

Notebook for the CS_Morgan Lab at CLEF 2023

Md. Rakibul Hasan [1], Oyebisi Layode [1] and Md Mahmudur Rahman [1]

[1] *Computer Science Department, Morgan State University, Baltimore, Maryland*

Abstract

This work discusses the participation of CS_Morgan in the Concept Detection and Caption Prediction tasks of the ImageCLEFmedical 2023 Caption benchmark evaluation campaign. The goal of this task is to automatically identify relevant concepts and their locations in images, as well as generate coherent captions for the images. The dataset used for this task is a subset of the extended Radiology Objects in Context (ROCO) dataset. The implementation approach employed by us involved the use of pre-trained Convolutional Neural Networks (CNNs), Vision Transformer (ViT), and Text-to-Text Transfer Transformer (T5) architectures. These models were leveraged to handle the different aspects of the tasks, such as concept detection and caption generation. In the Concept Detection task, the objective was to classify multiple concepts associated with each image. We utilized several deep learning architectures with 'sigmoid' activation to enable multilabel classification using the Keras framework. We submitted a total of five (5) runs for this task, and the best run achieved an F1 score of 0.4834, indicating its effectiveness in detecting relevant concepts in the images. For the Caption Prediction task, we successfully submitted eight (8) runs. Our approach involved combining the ViT and T5 models to generate captions for the images. For the caption prediction task, the ranking is based on the BERTScore, and our best run achieved a score of 0.5819 based on generating captions using the fine-tuned T5 model from keywords generated using the pre-trained ViT as the encoder.

Keywords

Medical Imaging, Image Annotation, Caption Prediction, Concept Detection, Multi-label Classification, Deep Learning, Natural Language Processing, GPT2, T5, MRCNN, ViT, Ensemble

1. Introduction

Generating accurate captions and concepts for biomedical images through automated processes poses a significant challenge in the field of artificial intelligence (AI). Successfully addressing this challenge necessitates the utilization of techniques from Computer Vision to comprehend the underlying concepts within the image, as well as the ability to grasp the relationships that exist among these concepts [1-2]. Additionally, techniques from natural language processing (NLP) are crucial for generating descriptive text that effectively represents the content of the image.

[1]CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece
EMAIL: mdhas1@morgan.edu (A. 1); oylay1@morgan.edu (A. 2); md.rahman@morgan.edu (A. 3)
ORCID: 0000-0002-6179-2238 (A. 1); 0000-0002-6924-0390 (A. 2); 0000-0003-0405-9088 (A. 3)

CEUR Workshop Proceedings (CEUR-WS.org)

Over the past decade, remarkable advancements have been achieved in leveraging Deep Learning methodologies to automatically generate clinical reports and captions based on medical images. The primary objective of these advancements is to support healthcare professionals in making precise decisions and enhance the efficiency of the diagnostic process [3].

By employing Computer Vision techniques, AI systems can interpret the complex visual elements and structures within biomedical images, enabling them to identify and understand the key concepts and features depicted. This involves tasks such as object recognition, segmentation, and image classification [1-3].

Moreover, the AI system must possess the capability to establish connections and associations among the identified concepts within the image. This contextual understanding is vital for generating coherent and meaningful descriptions [4-5]. Natural language processing techniques come into play here, as they facilitate the conversion of the interpreted concepts into human-readable text [4-5]. This process involves tasks such as language modeling, semantic analysis, and text generation [4-5].

The integration of these techniques from both Computer Vision and NLP has proven instrumental in automating the generation of captions and concepts for biomedical images [4-5]. This advancement in AI technology provides valuable support to clinicians by assisting them in making accurate and informed decisions in a timely manner [5]. By reducing the time required for diagnosis and enhancing the comprehensiveness of medical reports, these AI systems contribute to improving patient care and overall healthcare outcomes [5].

2. Dataset and Task Overview

The ImageCLEFmedical 2023 caption dataset [6-7] was divided into the training, validation, and test set. The training set contains 60,918 radiology images, the validation set contains 10,437 radiology images while the test set contains 10,473 radiology images. For the training and validation set, every image had a corresponding textual description (caption) that was provided.

The Concept Detection Task is an important step in achieving automatic image captioning and scene comprehension in the field of medical images [6]. Its purpose is to identify and locate relevant concepts within a large collection of medical images. By analyzing the visual content of the images, this subtask lays the groundwork for understanding the various elements that make up captions. Furthermore, these identified concepts can be utilized for context-based image retrieval and information retrieval. The evaluation of this task is performed using set coverage metrics, including precision, recall, and their combinations. These metrics measure the extent to which the set of concepts detected by the system aligns with the ground truth. In the Concept Detection Task, a subset of the UMLS 2022AB release was employed to generate the concepts. UMLS offers a range of tools and resources that empower users to access, comprehend, and navigate diverse health-related terminologies, including medical codes, classifications, ontologies, and databases. Within the biomedical domain, the UMLS employs a Concept Unique Identifier (CUI) as a specific identifier to represent distinct concepts. Each concept present in UMLS is assigned a CUI, which ensures a standardized and consistent means to reference and establish connections among related information across various terminologies and resources within the UMLS framework. ImageCLEF medical 2023 training dataset [6-7] has 227,156 labels/concepts and the validation dataset combinedly has 40,042 labels/concepts. Among these, 2,125 unique concept identifiers are used once or multiple times, because each image is labeled by one/multiple CUIs. Figure 1 illustrates the frequency of the most CUIs (both from train and valid dataset) and their corresponding medical terms.

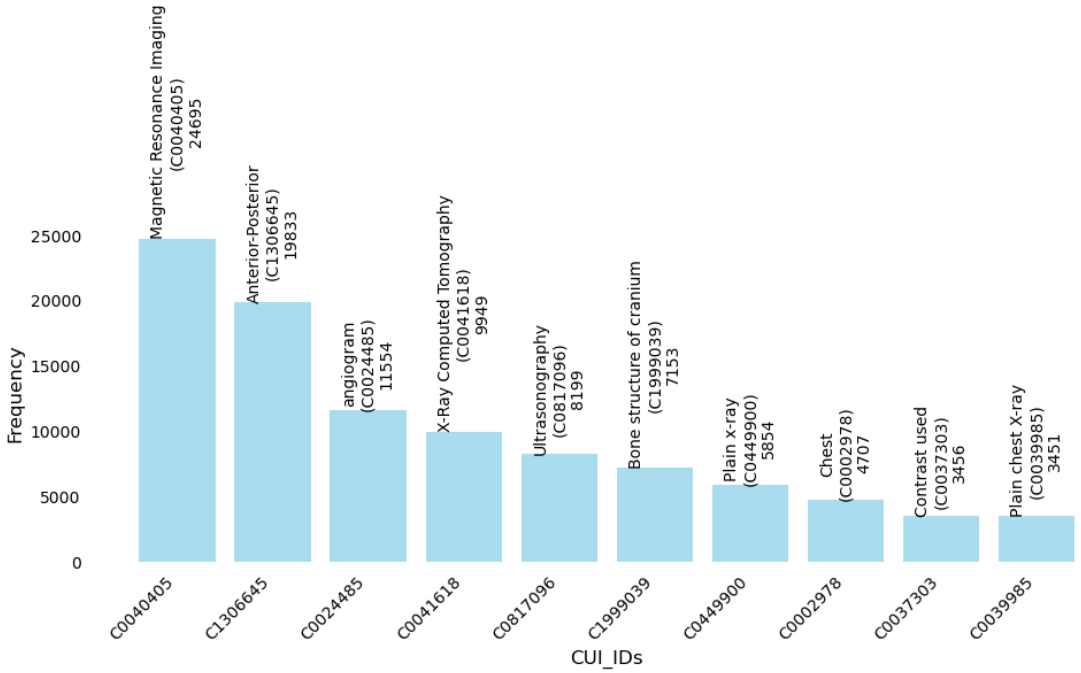

Figure 1: The most 10 frequent CUIs and their corresponding medical terms

On the other hand, when making a medical inference or captions on medical images, it is not uncouth to identify key parts of the image that can be combined to make a meaningful deduction on its possible contents [6-7]. This approach to deducing meaning from images draws some similarities with how visual transformers process image data. In recent times, transformers, which are based on the attention mechanism [8], were primarily utilized in sequence modeling and machine translation tasks. However, Transformers have gradually emerged as the predominant deep learning models for many NLP tasks [9]. Visual transformers in a similar manner to the application of Transformers in NLP tasks place importance on certain parts of the image features when making inference on an image. This is different from legacy convolutional neural networks (CNN) models that make inference on an image based on the entire image feature space. The mechanism of action of visual transformers suit image captioning tasks better than CNN models since the words in the caption typically only describe a part of the image. This paper involves the annotation of medical images to generate textual descriptions (captions) using the dataset under the ImageCLEFmed 2023 caption prediction task [6-7], which is a subset of a larger Radiology Objects in COntext (ROCO) dataset [10]. Our last participation involved the use of a visual transformer that consisted of an encoder-decoder architecture where meaningful image representations were generated from the encoder region, attention was then subsequently placed on these image representations to iteratively generate a sequence of words in the decoder. This year's participation builds on the previous work by replacing the trained decoder with a fine-tuned pretrained language model. The encoder was also replaced with pretrained models so the millions of mappings learned from the pretrained models can be leveraged for better results. This year, the evaluation criteria for the results obtained is given as the BERTScore between the generated captions and the ground truth captions.

3. Methods

We approached the Image Captioning and Concept detection tasks as two separate problems. The methods used in our participation are described below.

3.1. Medical Concept Detection Task

The task of concept detection is often viewed as a multi-label classification problem. This means that it involves predicting one or more concept labels for each instance of an image. In other words, an

image can be associated with multiple classes or concepts simultaneously, and these concepts are not mutually exclusive. The goal is to identify and assign the relevant concept labels to the image accurately, allowing for the possibility that an image may be associated with zero or more concepts. The approach implemented here for the concept detection task can be illustrated by addressing the following steps.

3.1.1. Data Preparation

The dataset used for caption prediction tasks consists of radiology images along with their corresponding labels or annotations. You can refer to Figure 3 and 4 for visual representations of this dataset. During the data preparation phase, both the images and labels undergo preprocessing. For all the submission runs, the image preprocessing techniques involve reshaping, rotation (at 45 or 90 degrees), and flipping (horizontally and vertically) augmentation techniques. Reshaping is performed to obtain two different image shapes, namely (224 x 224 x 3) and (331 x 331 x 3), as required by the pretrained Convolutional Neural Network (CNN) models used for the multi-label prediction task. This allows the images to be compatible with the CNN models and ensures consistent input dimensions for the network. As for the labels or annotations, they are transformed into a vectorized format using the Multi-label Binarizer method from the scikit-learn library. This method represents the multiple labels or Clinical Unique Identifiers (CUIs) as binary vectors. In this dataset, there are a total of 2,125 unique CUIs used in both the training and validation sets.

The steps of reshaping the images to match the requirements of the CNN models and converting the multi-label annotations into a binary representation using the Multi-label Binarizer method from scikit-learn ensure that the dataset is appropriately formatted and ready for use in the caption prediction tasks.

3.1.2. Model Preparation

To tackle the task of concept detection, we approached it by submitting five different attempts. For the first two runs, we utilized two pre-trained convolutional neural networks (CNNs) such as DenseNet169 [11] and DenseNet121 [11], which were trained on datasets like ImageNet [12] and CheXNet [13], respectively. On the fourth run we employed the Vision Transformer (ViT) [14] for image feature extraction followed by the k-nearest neighbors (kNN) algorithm to find the most similar training/validation images for a test image.

Furthermore, we employed an ensemble method based on the weighted average technique, using the ConvNeXtLarge [15] and NasNetLarge [16] CNN architectures on the third run. This involved combining the predictions from these models in a weighted manner. Additionally, we applied a majority voting technique-based ensemble method on several ImageNet weight based pre-trained CNNs including ResNet50 [17], Xception [18], VGG16 [19], and InceptionResNetV2 [20] on our fifth run.

For a more comprehensive understanding of these submission runs, you can refer to Section 4.1, which provides further details.

3.1.3. Training and Prediction

We adopted the Keras framework to carry out the training and prediction processes. All the models mentioned above were constructed by initializing the Adam optimizer and compiled using binary cross-entropy. This choice was made instead of using categorical cross-entropy to treat each output label as an independent Bernoulli distribution, allowing for the possibility that the labels are not mutually exclusive.

Once the training is completed, both the models and label binarizers are saved to a disk, specifically a cloud storage system, for future use. During the prediction phase on the test set, these saved models and binarizers are loaded to facilitate the prediction process.

To ensure efficient computation and handle the high memory requirements of the training process, we acquired a high-performance computing infrastructure equipped with four NVIDIA® T4 GPU

drivers (4-units) from a cloud service provider. These resources were utilized in parallel using the TensorFlow 2.11 mirrored strategy, which allows for distributed training across multiple GPUs.

Furthermore, the training procedures were conducted on the VertexAI workbench, a component of the Google Cloud platform specifically designed for machine learning tasks. This choice of platform provided a convenient and robust environment for performing the necessary computations and managing the overall training process.

3.2. Medical Image Captioning Task

The methods employed in the image captioning task majorly involved the use of pretrained models. The first approach involved training a Stochastic Gradient Descent (SGD) [21] classifier that will return the most similar captions for a query image. The SGD classifier was trained on bottleneck features obtained from two types of pretrained models, Mask Region with Convolutional Neural Networks (MRCNN) [22] pretrained on the MSCOCO dataset [23] and Google Vision Transformer (ViT) [14] trained on ImageNet. The second approach was divided into two stages with the first stage involving training a transformer model to sequentially generate keywords based on an image input. The MRCNN [22] and ViT [14] pretrained model acted as the encoder part of the transformer while a decoder was trained to sequentially generate keywords based on the image embeddings and the previously seen keywords. The second stage involved fine-tuning different types of pretrained language models to generate meaningful captions from the keywords generated from the first stage. Generative Pretrained Transformer 2 (GPT2) [24] and Google T5 [25] were fine-tuned for this purpose.

3.2.1. Text Preprocessing (Generating Keywords from Captions)

The ImageCLEFmedical 2023 caption training dataset [6-7] has a corpus of 23,237 words. An analysis of the distribution of these words shows that only about 295 words occur in more than 500 captions with there being 19,877 words that occur in less than 20 captions out of the entire 60,918 image captions. The keywords were selected as the 500 topmost occurring words excluding English stop words like “and”, “when”, “if”, “a”, “the” etc. (See Table 1 for some instances).

Table 1

Example of Captions and corresponding Keywords

#	Caption	Keywords
1	epicardial vessel from the distal right coronary artery white arrow collateralize the diagonal branch black arrow	vessel distal right coronary artery white arrow branch black arrow
2	plain xray of the patient before surgery show an extensive soft tissue swell	plain xray patient surgery extensive soft tissue
3	computed tomography show a renal transplant lie in the right iliac fossa and a polycystic leave native kidney arise from within the patient’s pelvis	tomography renal right iliac fossa leave kidney within pelvis

3.2.2. Image Feature Encoding

We chose MRCNN [22] and ViT [14] as our feature encoders to obtain region/object-based encodings of the images.

The MRCNN [22] architecture was developed as an object instance segmentation model that adopts a two-stage approach which includes a first stage that proposes candidate bounding boxes for detected objects (Region Proposal Networks, RPN) and a second stage that predicts a class, bounding box offset, and a binary mask for each region of interest (ROI). The region proposal features just before the classification layer was adopted as the image encodings. These features were obtained from a version of the MRCNN [22] architecture pretrained on the MSCOCO dataset [23]. The encoded 16 x 1024-dimension image embeddings are subsequently passed into other parts of the image captioning model.

The ViT [14] architecture converts 16x16 patches of an image input into flattened linear projects that are combined with the positional embeddings and passed as a sequence of patch embeddings to an encoder. A classification multi-layer perceptron is placed on top of the encoder to map the encoded embeddings to predicted classes. 1,024-dimension features were obtained from the pooling layer just before the classification layer of the pre-trained ViT [14] model. These features represent the image embeddings.

3.2.3. SGD classifier + pre-trained features

The SGD classifier was trained using the scikit's library [26]. This involved training a Multioutput Classifier using an SGD classifier wrapped in a OneVsRest Classifier as the estimator. The classifier was also trained with scikit's adaptive learning rate option alongside the default options for the classifier. The same training parameters were applied to train the classifier on the MRCNN [22] and ViT [14] features. The results were generated from the classifier by obtaining the encoded features from the pretrained MRCNN [22] and ViT [14] models and querying the features on the classifier to return a caption that is most similar based on the query feature.

3.2.4. Keyword Transformer Model (KTM)

The transformer model used in this task consists of an encoder and decoder model that receives a 224 x 224 image alongside a tokenized text input padded up to a maximum length of 20 tokens including a start and end token.

3.2.4.1. Encoder-Decoder

The encoder model consists of a pretrained model that receives a 224 x 224 x 3 image input and gives bottleneck feature output of 16 x 1,024 for the MRCNN [22] model and 1,024 for the ViT [14] model.

The decoder model learns to map the image embedding output of the encoder model to text embeddings. The text embeddings are an addition of the text tokens (which are the indexes of the caption word in the corpus) passed to a Keras text embedding layer and the positional encodings of the words in the generated captions. The decoder auto-regressively tracks the positions of the predicted words as the encoder output is passed as the keys and value to the decoder while the prior decoder predictions are passed as the decoder query for which a next word is predicted.

The decoder architecture consists of an input layer (receives the encoder and text tokens) => the text tokens are passed to a text embedding layer (text embeddings + positional encodings) => the result of the text embedding layer is passed to a dropout layer => the result of the dropout layer and the image encoding output is passed to a decoder layer which consists of a causal self-attention followed by a cross attention layer, the text encodings are passed to the self-attention layer learn the relations between the words. The result of the self-attention is then passed to the cross-attention layer alongside the encoder output => the result of the cross-attention layer is then passed to a feed forward layer => the output of the decoder layer is passed to a dense layer with the number of neurons equivalent to the vocabulary size of the corpus. Figures 2 and 3 illustrate the corresponding decoder architectures.

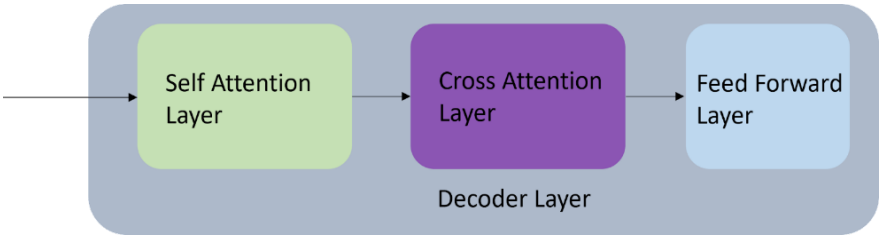

Figure 2: Image of the decoder layer used in the described model.

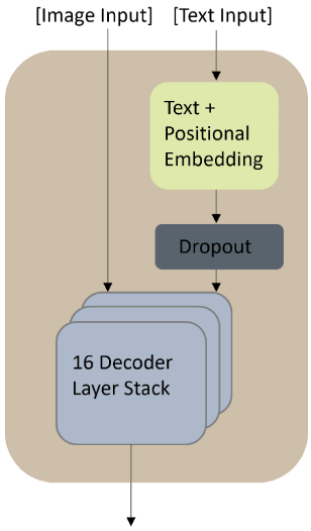

Figure 3: Image of the decoder model used in the described model.

3.2.4.2. Training

All the images from the ImageCLEFmedical 2023 caption dataset [6-7] were resized to 224 x 224 x 3 and normalized by dividing the pixel values by 255. The KTM was trained with an Adam optimizer [27] with a masked version of the sparse categorical entropy loss, a learning rate set at 0.001 and a batch size of 256. The decoder model was trained using a 256-feature projection dimension and a feed forward dimension of 256. The decoder was also trained on 8 attention heads and a stack of 16 decoder layers. The training was set for 200 epochs with early stopping applied and the last best model restored after a patience of 10.

3.2.5. Keyword to Caption Model (KTC)

The KTC model involved fine-tuning pretrained language models to generate meaningful caption sentences from keywords. GPT2 [24] and T5 [25] were fine-tuned on the ImageCLEFmedical 2023 caption dataset [6-7]. GPT-2 [24] is a transformers model that was pre-trained on a very large corpus of English data in a self-supervised manner [28]. The model largely follows after the OpenAI GPT model architecture [29] with few modifications including the layer normalization being moved to the input of each sub-block, an additional layer normalization being added after the final self-attention block, scaling the weights of residual layers at initialization by a factor of $1/\sqrt{N}$ where N is the number of residual layers, an expansion of the vocabulary to 50,257, an increase in the context size from 512 to 1,024 tokens [24]. The T5 [25] architecture follows the general transformer approach as originally proposed [8]. This involves mapping an input sequence of tokens to a sequence of embeddings that is subsequently passed into the encoder, with the encoder consisting of a stack of "blocks", each of which comprises a self-attention layer and a small feed-forward network. Layer normalization [30] is applied to the input of each subcomponent while a residual skip connection is used to add each subcomponent's input to its output. Dropout [31] is applied within the feed-forward network, on the skip connection, on the attention weights, and at the input and output of the entire stack. The T5 [25] decoder has a similar structure to the encoder except that on top of the self-attention layer that attends to the output of the encoder [25].

3.2.5.1. Fine-Tuning

The keywords were passed to the models as input while the captions were passed as the expected output. Both the input and output data were tokenized and padded, the input data was padded up to a maximum length of 20, while the output data was padded up to a maximum length of 50. GPT2 [24]

was fine-tuned for 25 epochs with an Adam weight decay optimizer [27], a learning rate of 0.0005 and a weight decay rate of 0.01. T5 [25] was fine-tuned for 41 epochs with a Adafactor optimizer, a learning rate of 0.0001, decay rate of 0.8 and a clip threshold of 1.0.

4. Description of Runs and Results

We submitted 13 runs in total for this year's participation, 8 runs were submitted for the caption prediction task while 5 runs were submitted for the concept detection task as shown in Table 2 and 3. For Concept Detection, the primary evaluation metric was the F1 score, we obtained a best score of 0.4834 for Run_2 and ranked 6th out of 9 participants. For the Caption Prediction task, the primary evaluation metric is the BERTScore, additional evaluation metrics used include ROUGE, METEOR, CIDEr, the BLEU score. The best score recorded was for Run_10 with a BERTScore of 0.2549. The best scores are in boldface. Overall, we ranked 6th out of 9 groups for the Concept Detection task and ranked 9th out of 13 groups for the Caption Prediction task.

4.1. Medical Concept Detection Task

Table 2 shows the summary of the run submissions for the concept detection task and respective F1 scores.

Table 2
List of the Run Submissions with the corresponding F1-scores**

Submissions	Architecture	Pretrained Weights	F1 Score*	F1-Score Manual*
Run_1	DenseNet 169	Imagenet	0.4368	0.8542
Run_2	**DenseNet 121**	**CheXNet**	**0.4834**	**0.8901**
Run_3	NasNetLarge and ConvNeXtLarge with Weighted Avg. Ensemble	Imagenet	0.0060	0.1444
Run_4	ViT (Feature Extraction) and kNN (k=14)	Imagenet	0.1008	0.3728
Run_5	ResNet50, Xception, VGG16, and InceptionResNetV2 with Ensemble/Late Fusion (Majority Voting Technique)	Imagenet	0.4791	0.8582

*The F1 Scores are found from https://fh-dortmund.sciebo.de/s/ZdPRXcPfMZ5cbUy
** Source: https://www.imageclef.org/2023/fusion

The submitted runs for the Concept Detect tasks are described as follows:

1. **Run_1_Dense169:** Run 1 corresponds to a model that uses the DenseNet169 [11] architecture. DenseNet169 [11] is a convolutional neural network architecture that has shown promising performance in various computer vision tasks, including multi-label classification of medical images [6, 7]. It is characterized by its densely connected layers, where each layer is directly connected to every other layer in a feed-forward fashion. When using DenseNet169 [11] for multi-label prediction in Keras, it is common to leverage pretraining on the Imagenet [12] dataset. Imagenet [12] is a large-scale dataset that consists of millions of labeled images from thousands of categories [12]. By utilizing transfer learning and fine-tuning techniques, we have adapted the pretrained DenseNet169 [11] model to our concept detection task using Keras libraries. A Dense layer with 2,125 unique number classes is used on the top followed by a flatten layer and global average pooling layer.

2. **Run_2_Dense121:** Run 2 corresponds to a model that uses the pre-trained DenseNet121 [11] architecture with CheXNet [13] weights. The original paper of CheXNet [13] trained on ChestX-ray14 dataset [13] is an architecture of 121 CNN layers to detect labels for various pathologies based on more than 100,000 chest X-rays images. This fine-tuned model shows promising results among the five submissions we made for the task. Fig. 4 illustrates our implemented model for this run.

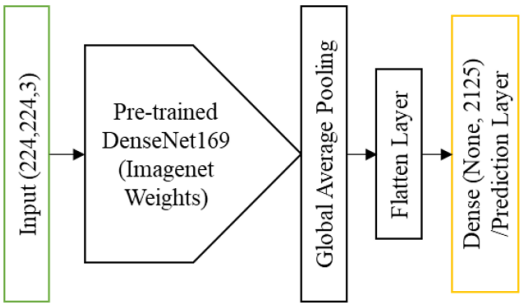

Figure 4: Architecture of Run_2_Dense121 (CheXNet)

3. **Run_3_Ensemble_Convnext_Nasnet:** Run 3 corresponds to a model that uses an ensemble of architectures, namely NasNetLarge [16] and ConvNeXtLarge [15], with a weighted average approach (Fig. 5).

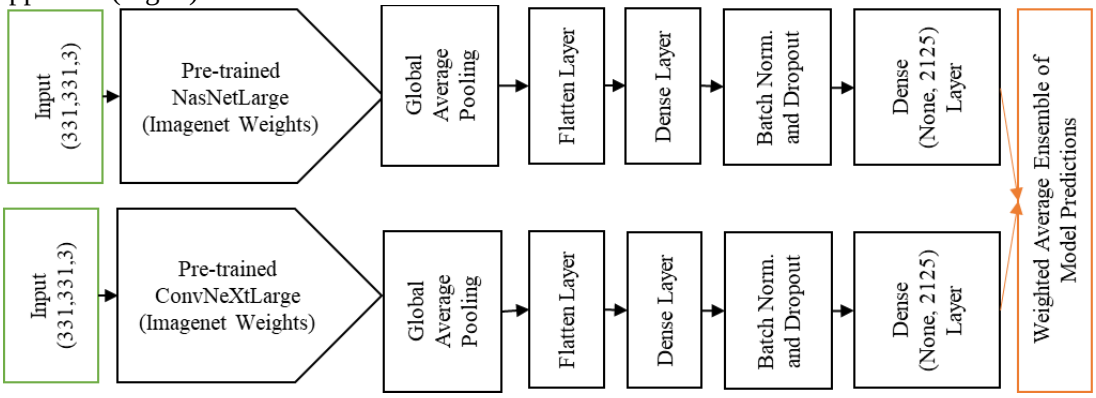

Figure 5: Architecture of Run_3_Ensemble_Convnext_Nasnet

NasNetLarge is a neural architecture search (NAS) network that automatically searches for an optimal CNN architecture [16]. It utilizes a reinforcement learning-based approach to discover an architecture that is well-suited for a given task. On the other hand, ConvNeXtLarge is another CNN architecture that focuses on capturing complex patterns and relationships within images [15]. It employs grouped convolutions and split-transform-merge operations to enhance its representation power and performance [15]. We implemented weighted average ensemble technique to combine predictions from these two architectures. Each model's predictions were weighted based on their validation accuracy. The purpose of the weighted average ensemble was to leverage the strengths of these models.

4. **Run_4_ViT_kNN:** Run 4 corresponds to a model that uses the pre-trained (ImageNet) Vision Transformer (ViT) architecture for feature extraction, combined with a k-nearest neighbors (kNN) approach with k=14. By leveraging self-attention mechanisms, ViT can capture global context and long-range dependencies within an image, leading to effective feature extraction and image understanding [14]. On the other hand, k-Nearest Neighbors (kNN) assigns labels to test samples based on the labels of the k most similar training and validation samples. In this case, kNN is applied to the features extracted by ViT. By finding the k nearest training images based on their feature similarity, the algorithm utilizes their known labels as the predicted labels for the test image. The value of k=14 has been determined through a process of experimentation and evaluation. The F1 scores for 100 different random values of k, ranging from 5 to 500, have

been calculated and compared to identify the value that yields the best performance. This process ensures that the chosen value of k provides the optimal balance between capturing the diversity of similar images and minimizing potential noise or outliers. Figure 6 depicts the architecture used on this fourth run.

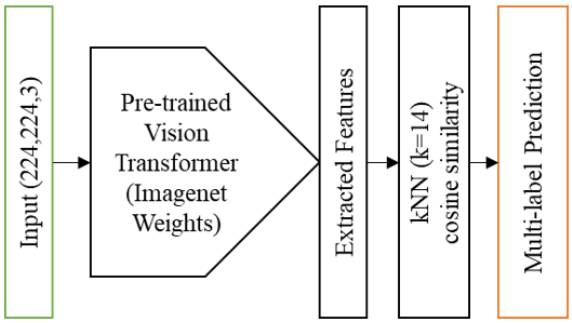

Figure 6: Architecture of Run_4_ViT_kNN

5. **Run_5_Ensemble_ResNet_VGG_Xcep_Incept**: Run 5 corresponds to a model that uses an ensemble (majority voting technique based) of multiple architectures (Fig. 7), including ImageNet [12] weight based pre-trained ResNet50 [17], Xception [18], VGG16 [19], and InceptionResNetV2 [20]. The ensemble approach utilizes a majority voting technique for decision-making. The ResNet50 architecture consists of residual blocks with skip connections, allowing the network to learn more complex representations. Xception is an extension of the Inception architecture that replaces the traditional convolutional layers with depth wise separable convolutions. VGG16 is a deep convolutional neural network architecture with 16 layers. Ensemble methods combine the predictions of multiple models to improve overall performance and robustness. The majority voting technique is a simple ensemble approach where the final prediction is determined by selecting the label that receives the most votes from individual models. See Figure 7 for further illustration of the architecture.

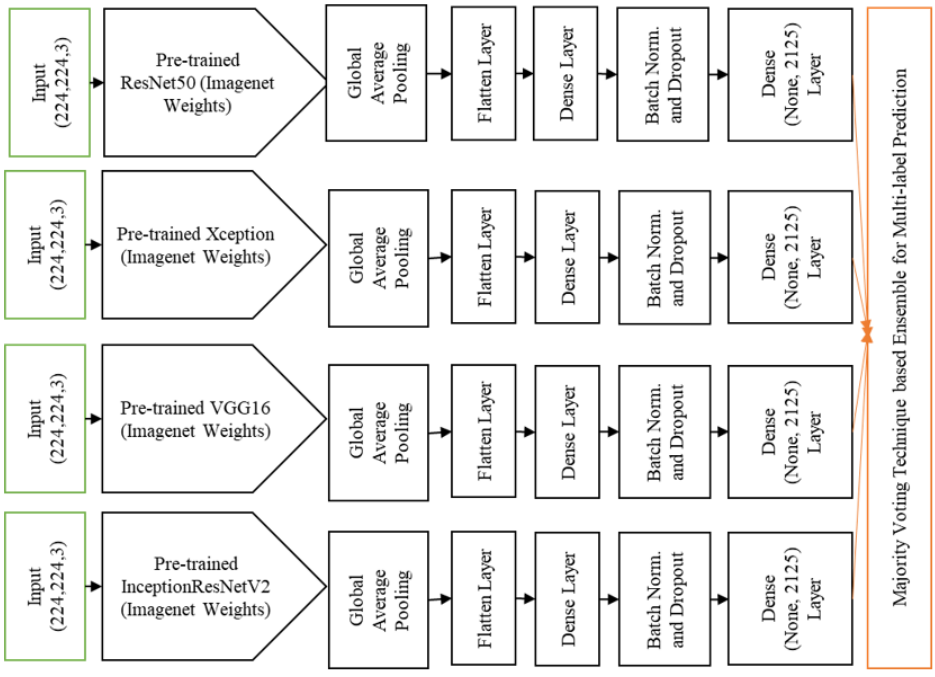

Figure 7: Architecture of Run_5_Ensemble_ResNet_VGG_Xcep_Incept

4.2. Medical Image Captioning Task

The submitted successful runs for the Caption Prediction tasks are described as follows:

1. **Run_4:** Run 4 involved using the trained KNN classifier used to retrieve the most similar caption based on extracted features from the pre-trained ViT [8] model.
2. **Run_5:** Run_5 involved using the trained KNN classifier used to retrieve the most similar caption based on extracted features from the pre-trained MRCNN [10] model.
3. **Run_6:** Run_6 involved using the trained transformer model described in section 2.2.4 to generate keywords using the pre-trained ViT [8] as the encoder.
4. **Run_8:** Run_8 involved using the trained transformer model described in section 2.2.4 to generate keywords using the pre-trained MRCNN [10] as the encoder.
5. **Run_9:** Run_9 involved generating captions using the fine-tuned T5 [12] model from keywords generated in Run_8.
6. **Run_10:** Run_10 involved generating captions using the fine-tuned T5 [12] model from keywords generated in Run_6.
7. **Run_12:** Run_12 involved generating captions using the fine-tuned GPT2 [11] model from the keywords generated in Run_6.
8. **Run_13:** Run_13 involved generating captions using the fine-tuned GPT [11] model from the keywords generated in Run_8.

Table 3 shows the summary of the run submissions for the caption prediction task and the respective BERTScore, ROUGE, BLEURT, BLEU, METEOR, CIDEr, and CLIP Score values. Our best scores and explanations for the corresponding performance metrices are as follows**.**

- **BERTScore = 0.5819:** BERTScore or Bidirectional Encoder Representations from Transformers measures the similarity between machine-generated text and reference text based on contextual embeddings. The BERTScore ranges from 0 to 1, with higher scores indicating greater similarity or better quality. In this case, a BERTScore of 0.5819 suggests that the machine-generated text has a moderate level of similarity to the reference text.
- **ROUGE = 0.1564:** ROUGE or Recall-Oriented Understudy for Gisting Evaluation score indicates the level of overlap between machine-generated captions and reference captions, as measured by the ROUGE evaluation metrics (e.g., ROUGE-N (n-gram overlap), ROUGE-L (longest common subsequence), and ROUGE-S (skip-bigram co-occurrence)). ROUGE scores typically range from 0 to 1, with higher scores indicating greater similarity or better quality. The score of 0.1564 suggests a relatively low level of overlap between the machine-generated output and the reference text.
- **BLEURT = 0.2649:** BLEURT or Bilingual Evaluation Understudy with Representations from Transformers score of 0.2649 indicates the moderate level of quality or similarity of a machine-generated translation compared to human reference translations, as measured by the BLEURT evaluation metric. BLEURT scores typically range from negative values to positive values, with higher positive scores indicating better quality or similarity. This metric utilizes a pre-trained transformer-based model to calculate the similarity score.
- **BLEU = 0.1331:** BLEU or Bilingual Evaluation Understudy is an evaluation metric used to measure the quality of machine translations. BLEU scores range from 0 to 1, with higher scores indicating better translation quality. The score of 0.1331 suggests a relatively low level of translation quality. Moreover, it measures the precision of n-grams (commonly unigrams, bigrams, trigrams, etc.) in the machine-generated translation compared to the reference translations.
- **METEOR = 0.0568:** METEOR or Metric for Evaluation of Translation with Explicit Ordering score of 0.0568 is an evaluation metric used to measure the quality of machine translations. METEOR scores typically range from 0 to 1, with higher scores indicating better translation quality. METEOR considers both lexical and syntactic aspects of translation quality.
- **CIDEr = 0.1731:** CIDEr or Consensus-based Image Description Evaluation is used in image captioning tasks to assess the quality of generated captions. CIDEr scores typically range from 0 to 1, with higher scores indicating better caption quality. CIDEr considers the consensus among reference captions and encourages diversity in the generated captions. Here, the score represents a relatively low level of caption quality.

- **CLIP Score = 0.7771**: CLIP (Contrastive Language-Image Pretraining) is a method developed by OpenAI to measure the proximity of encoded text and image vectors. The CLIP score-based Cosine similarity ranges from +1 to -1. Here the calculated score of 0.7771 represents moderate identical vectors.

Table 3
List of the Run Submissions ranked by corresponding BERTScores**

Runs	BERTScore*	ROUGE*	BLEURT*	BLEU*	METEOR*	CIDEr*	CLIP Score*
Run_4	0.5791	0.1541	**0.2649**	**0.1331**	**0.0568**	**0.1731**	**0.7771**
Run_5	0.5508	0.1070	0.2373	0.1040	0.0351	0.0481	0.7165
Run_6	0.5438	0.1107	0.1817	0.0026	0.0329	0.0925	0.7582
Run_8	0.5087	0.0264	0.1205	0.0034	0.0107	0.0125	0.6819
Run_9	0.5419	0.0924	0.1735	0.0406	0.0210	0.0187	0.6821
Run_10	**0.5819**	**0.1564**	0.2242	0.0566	0.0436	0.0840	0.7593
Run_12	0.5558	0.1272	0.2569	0.1199	0.0344	0.0164	0.7338
Run_13	0.5481	0.1144	0.2435	0.1180	0.0323	0.0142	0.6910

*The scores are found from https://fh-dortmund.sciebo.de/s/ZdPRXcPfMZ5cbUy
** Source: https://www.imageclef.org/2023/fusion

5. Conclusion

This working note paper describes the approaches and outcomes of the CS_Morgan group's involvement in the ImageCLEFmedical 2023 Caption task. We participated in both the Concept Detection and Caption Prediction tasks within this evaluation. Our most successful results were obtained by employing a Convolutional Neural Network (CNN) architecture for concept detection and transformer-based methods, specifically ViT and T5, for the caption prediction task.

Looking ahead, we intend to further investigate Transformers that are tailored to the medical domain and utilize improved fusion mechanisms. This area of research has gained significant interest in the medical field as it enables the capture of global context, surpassing the capabilities of CNNs, which primarily focus on local receptive fields. We believe that exploring these domain-specific Transformers will enhance our future performance in the ImageCLEFmedical task.

Acknowledgements

This work is supported by the National Science Foundation (NSF) grant (ID. 2131307) “CISE-MSI: DP: IIS: III: Deep Learning-Based Automated Concept and Caption Generation of Medical Images Towards Developing an Effective Decision Support."

References

[1] Sanjay Subramanian, Lucy Lu Wang, Sachin Mehta, Ben Bogin, Madeleine van Zuylen, Sravanthi Parasa, Sameer Singh, Matt Gardner, and Hannaneh Hajishirzi. Medicat: A dataset of medical images, captions, and textual references, 2020.

[2] Djamila-Romaissa Beddiar, Mourad Oussalah, and Tapio Sepp ̈anen. Automatic captioning for medical imaging (mic): a rapid review of literature. *Artificial Intelligence Review*, pages 1–58, 2022.

[3] Dina Demner-Fushman, Sameer Antani, Matthew Simpson, and George R Thoma. Design and development of a multimodal biomedical information retrieval system. *Journal of Computing Science and Engineering*, 6(2):168–177, 2012.

[4] John Pavlopoulos, Vasiliki Kougia, and Ion Androutsopoulos. A survey on biomedical image captioning. In *Proceedings of the second workshop on shortcomings in vision and language*, pages 26–36, 2019.

[5] Songhua Xu, Jamie McCusker, and Michael Krauthammer. Yale image finder (yif): a new search engine for retrieving biomedical images. *Bioinformatics*, 24(17):1968–1970, 2008.

[6] Bogdan Ionescu, Henning Müller, Ana-Maria Drăgulinescu, Wen-wai Yim, Asma Ben Abacha, Neal Snider, Griffin Adams, Meliha Yetisgen, Johannes Rückert, Alba García Seco de Herrera, Christoph M. Friedrich, Louise Bloch, Raphael Brüngel, Ahmad Idrissi-Yaghir, Henning Schäfer, Steven A. Hicks, Michael A. Riegler, Vajira Thambawita, Andrea Storås, Pål Halvorsen, Nikolaos Papachrysos, Johanna Schöler, Debesh Jha, Alexandra-Georgiana Andrei, Ahmedkhan Radzhabov, Ioan Coman, Vassili Kovalev, Alexandru Stan, George Ioannidis, Hugo Manguinhas, Liviu-Daniel Ștefan, Mihai Gabriel Constantin, Mihai Dogariu, Jérôme Deshayes, Adrian Popescu, Overview of the ImageCLEF 2023: Multimedia Retrieval in Medical, Social Media and Recommender Systems Applications, in Experimental IR Meets Multilinguality, Multimodality, and Interaction.Proceedings of the 14th International Conference of the CLEF Association (CLEF 2023), Springer Lecture Notes in Computer Science LNCS, Thessaloniki, Greece, September 18-21, 2023.

[7] Johannes Rückert, Asma Ben Abacha, Alba G. Seco de Herrera, Louise Bloch, Raphael Brüngel, Ahmad Idrissi-Yaghir, Henning Schäfer, Henning Müller and Christoph M. Friedrich. Overview of ImageCLEFmedical 2023 – Caption Prediction and Concept Detection, in Experimental IR Meets Multilinguality, Multimodality, and Interaction. CEUR Workshop Proceedings (CEUR-WS.org), Thessaloniki, Greece, September 18-21, 2023.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[9] Yang Liu, Yao Zhang, Yixin Wang, Feng Hou, Jin Yuan, Jiang Tian, Yang Zhang, Zhongchao Shi, Jianping Fan, and Zhiqiang He. A survey of visual transformers. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[10] Obioma Pelka, Sven Koitka, Johannes R ̈uckert, Felix Nensa, and Christoph M Friedrich. 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: *7th Joint International Workshop, CVII-STENT 2018 and Third International Workshop, LABELS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings 3*, pages 180–189. Springer, 2018.

[11] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.

[13] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[15] Saining Xie, Ross Girshick, Piotr Doll´ar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[16] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Fran¸cois Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[20] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[21] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[22] Kaiming He, Georgia Gkioxari, Piotr Doll´ar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll´ar, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference*, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pages 740–755. Springer, 2014.

[24] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[25] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[26] Fabian Pedregosa, Ga¨el Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. the *Journal of machine Learning research*, 12:2825–2830, 2011.

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[28] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R´emi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[29] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[30] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[31] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. The *journal of machine learning research*, 15(1):1929–1958, 2014.