AUEB/DMST at ImageCLEFmedical Caption 2026

Notebook for AUEB/DMST at CLEF 2026

Giorgos Karandreas[1], Panos Louridas[1]

[1]Department of Management Science and Technology, Athens University of Economics and Business, Athens, Greece

Abstract

This paper presents the AUEB/DMST submission to the 2026 ImageCLEFmedical Caption Tasks, namely the Concept Detection and Caption Prediction Tasks. The Concept Detection Task involved identifying the presence of relevant UMLS (Unified Medical Language System) concepts in a large corpus of radiology images. The Caption Prediction Task involved generating coherent medical captions describing full radiology images. This year, two additional synthetic variants of these tasks were introduced, sharing the same objectives but using synthetic data. For the Concept Detection Tasks, we employed a weighted multi-model soft-voting ensemble of convolutional neural network (CNN) and Vision Transformer (ViT) architectures. For the Caption Prediction Task, we implemented a captioning architecture with dual visual encoders and a single decoder, connected by a Q-Former-based cross-attention bridge. Overall, these approaches ranked 1st place in the Synthetic Concept Detection Task, 2nd in the Synthetic Caption Prediction Task, 4th in the Standard Caption Prediction Task, and 5th in the Standard Concept Detection Task.

Keywords

ImageCLEFmedical, Medical Image Captioning, UMLS Concept Detection, Vision Transformers, Convolutional Neural Networks, Multi-Label Classification, Vision-Language Models

1. Introduction

Interpreting and summarizing the insights gained from medical images such as radiology output is a time-consuming task that involves highly trained experts and often represents a bottleneck in clinical diagnosis pipelines. Consequently, there is a considerable need for automatic methods that can approximate this mapping from visual information to condensed textual descriptions. The more image characteristics are known, the more information can be coupled to the radiology scans, making radiologists more efficient at interpreting them.

The need has motivated shared benchmarks such as ImageCLEFmedical Caption. ImageCLEF [1] is an evaluation initiative launched in 2003 under the Cross Language Evaluation Forum (CLEF) aimed at the advancement of visual media analysis, indexing, classification, and retrieval technologies for multimodal data. Under the umbrella of ImageCLEF, ImageCLEFmedical is a specialized set of evaluation tasks that benchmark machine learning systems for medical image analysis. Among these tasks, the Caption Task is particularly relevant to the aforementioned problem of improving efficiency in radiology diagnostics.

1.1. Task Overview

This year marks the 10th edition of the Caption Task [2], whose objective is the automatic interpretation of radiology images by linking visual medical content with clinically relevant textual descriptions. To address the different aspects of medical image captioning, the task is organized into multiple subtasks, namely the Concept Detection task, the Caption Prediction task, and the Explainability task, each of which is described in more detail below.

CLEF 2026 Working Notes, 21–24 September 2026, Jena, Germany

georkarandreas@gmail.com (G. Karandreas); louridas@aueb.gr (P. Louridas)

0000-0002-3971-4612 (P. Louridas)

1.1.1. Concept Detection Task

In the Concept Detection subtask, given a radiology image, a system must predict the set of medically relevant UMLS Concept Unique Identifiers (CUIs) [3] associated with that image. These concepts can be interpreted as labels with semantic value that capture medical entities that are reflected in the image's caption. In machine learning terms, the input is a single radiology image and the output is an unordered, unfixed-size label set, so the task can be modeled as multi-label image classification over a fixed concept vocabulary, with independent sigmoid probability outputs for each label.

The evaluation methodology is based on image-level F1. For each image, the predicted concept set and the ground-truth concept set are represented as binary vectors indicating concept presence, and a F1 score for label 1 (binary F1) is computed for that image. These image-level F1 values are then averaged over the test set. The primary score uses all concepts. The secondary score repeats the same computation after filtering to a manually curated concept subset used by the organizers.

1.1.2. Caption Prediction Task

In the Caption Prediction subtask, systems must produce a coherent caption for each radiology image using the visual content and optionally the UMLS concept vocabulary identified in the Concept Detection subtask. In machine learning terms, the task takes a radiology image as input and outputs an ordered natural-language sequence, so it is best viewed as an image-conditioned text-generation problem.

The evaluation methodology uses six metrics covering caption relevance and factuality. The relevance metrics are image-caption similarity, BERTScore Recall with inverse document frequency (IDF) weighting, ROUGE-1 F-measure, and BLEURT. The factuality metrics are UMLS Concept F1, computed using MedCAT-based concept extraction, and AlignScore. The final participant ranking is determined by the average score over all six metrics.

Note. The official task evaluation scripts are publicly available at https://github.com/taubsity/clef-caption-evaluation-2026.

1.1.3. Explainability Task

The Explainability subtask focuses on the human interpretability of medical image captioning systems, aiming to provide explanations for the model predictions. However, our participation was limited to the Concept Detection and Caption Prediction subtasks. Therefore, no methods, submissions, or results are reported for the Explainability subtask in this paper.

1.1.4. Synthetic Tasks

For its 10th edition, the ImageCLEFmedical Caption task introduced two synthetic data variants of the standard Concept Detection and Caption Prediction subtasks: Concept Detection Synthetic and Caption Prediction Synthetic. For these subtasks, new captions were generated and new concepts were derived from those captions. We participated in both the standard and synthetic data variants.

The synthetic data subtasks follow the same task definitions, structure and objective as their standard counterparts, with the exception that the Synthetic Concept Detection subtask does not include a secondary evaluation score. For this reason, we do not describe them separately. Unless otherwise stated, the descriptions of Concept Detection and Caption Prediction, as well as the methodology presented later in this paper, apply to both the standard and synthetic data variants.

1.2. Our Participation

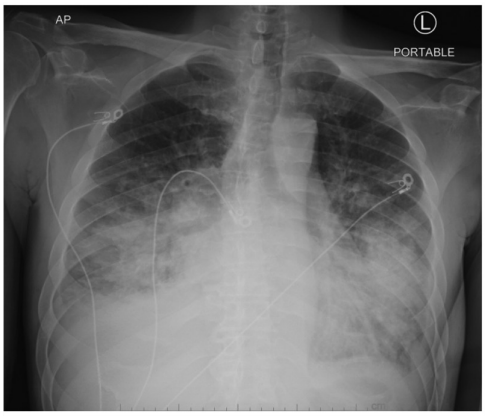
In this paper, we describe our submissions to the ImageCLEFmedical Caption task. Our work focuses on both the standard and synthetic Concept Detection and Caption Prediction subtasks. For Concept Detection, we treat the task as a multi-label image classification problem and explore a weighted

soft-voting ensemble of convolutional neural network (CNN) and vision transformer (ViT) models for predicting UMLS concept labels. For Caption Prediction, we investigate a dual-visual-encoder, single-decoder architecture that integrates a Querying Transformer bridge between the visual encoders and the language decoder. Overall, our submitted systems ranked 1st in Synthetic Concept Detection, 2nd in Synthetic Caption Prediction, 4th in Standard Caption Prediction, and 5th in Standard Concept Detection. The paper reports the methodology, implementation details, and evaluation results of our submitted systems.

2. Data

This year's ImageCLEFmedical Caption Task uses an updated and extended version of the ROCOv2 multimodal radiology image dataset [4]. ROCOv2 is itself an updated version of the original ROCO dataset published in 2018 [5]. It is composed of radiological images and their associated medical concepts and captions sourced from biomedical articles of the PubMed Central Open Access (PMC OA) subset. The dataset covers multiple radiological imaging modalities, including X-ray, computed tomography (CT), magnetic resonance imaging (MRI), ultrasound, positron emission tomography (PET), angiography, and combined imaging modalities.

While the public ROCOv2 release contains 79,789 radiology images split into training, validation, and testing datasets [4], the ImageCLEFmedical Caption task uses challenge-specific releases rather than reusing the ROCOv2 splits. This distinction is important because individual challenge editions may extend or repartition the data. Both the 2024 and 2025 editions already used larger releases, and the 2026 edition follows the same pattern. Accordingly, all results reported in this paper are based on the official 2026 ImageCLEFmedical Caption splits.



ID: ImageCLEFmedical_Caption_2026_train_3335

CUI	**UMLS Term**
C1306645	Plain x-ray
C0817096	Chests
C0013687	Effusion
Caption	
Chest X-ray above shows central vascular prominence with abnormal alveolar opacities in the mid and lower lungs bilaterally in addition to small effusions.	

Figure 1: Representative sample from the dataset, showing the input image and its corresponding annotation.

2.1. Standard Data

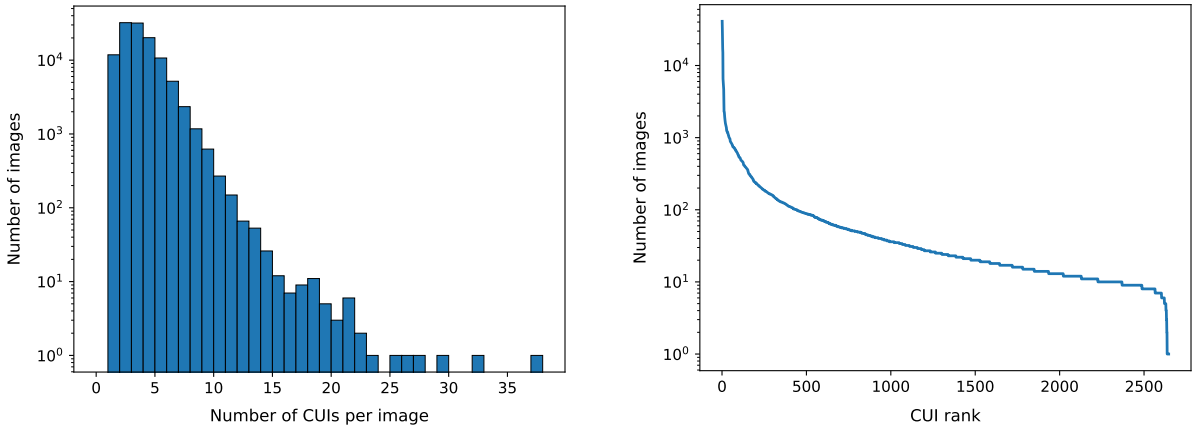
Both standard subtasks share the same image collection, but use different associated annotations. In this year's edition, the development set consists of 116,604 images, split into 97,364 training images and 19,240 validation images. Each development image is annotated with a set of medical concepts expressed as UMLS CUIs for the Concept Detection task and one diagnostic caption for the Caption Prediction task. The test set consists of 15,249 previously unseen images, for which only the images were released to participants. In total, 131,853 images were used across all splits.

The CUIs were generated using a reduced subset of the UMLS 2022AB release. According to the official ImageCLEFmedical Caption 2026 task description, concepts were filtered by semantic type to improve the feasibility of recognizing them from the images, and low-frequency concepts were removed based on recommendations from previous editions of the task [2].

Figure 1 shows a representative image from the dataset together with its associated annotations.

2.1.1. Concept Detection Data Statistics

To better understand the CUI label space, we analyzed the distribution of concepts in the official development set. The set contains 2,646 unique CUIs. On average, each image is associated with 3.22 concepts, with a mode of 2 concepts per image. The number of concepts per image ranges from 1 to 37. Figure 2a shows the distribution of the number of CUIs assigned to each image.



(a) Distribution of CUI counts per image.

(b) Long-tailed CUI frequency distribution.

Figure 2: Standard Concept Detection label statistics. Both y-axes use logarithmic scale to improve visibility of less frequent values.

The concept-label distribution follows a clear long-tailed pattern, as shown in Figure 2b. The distribution is consistent with observations from the previous year [6]. A small set of frequent CUIs accounts for a large portion of the annotations, with the most common concepts appearing in a third of the development images. In contrast, the majority of CUIs occur in only a small percentage of images, with a global mean occurrence rate of 0.12% (≈142 images per CUI). This creates a challenging label space in which models must learn both common visual concepts and rare medical labels from limited examples. Table 1 reports the most frequent CUIs and their occurrence rates, which include information related to imaging modality (predominantly X-rays) and body regions.

In addition to the full concept vocabulary, the standard Concept Detection data include a manually annotated subset of concepts used by the organizers for the secondary evaluation score. In ROCOv2, this subset mainly covers basal radiological concepts, such as imaging modality and, for X-ray images, body region and projection directionality [4].

Table 1
Top 10 most frequent CUIs in the standard development data.

Rank	Images	Occurrence%	CUI	UMLS Term
1	40,992	35.15%	C0040405	Computed X Ray Tomography
2	31,818	27.29%	C1306645	Plain x-ray
3	18,587	15.94%	C0024485	Magnetic resonance imaging
4	17,143	14.70%	C0041618	Ultrasonography, NOS
5	15,066	12.92%	C0817096	Chests
6	6,537	5.61%	C0000726	Abdomens
7	6,055	5.19%	C0002978	Angiogram
8	5,637	4.83%	C0037303	Bone structure of cranium
9	5,342	4.58%	C0030797	Pelvis
10	4,662	4.00%	C0023216	Lower extremity

2.1.2. Caption Prediction Data Statistics

For the Caption Prediction subtask, we analyzed the caption annotations in the official development split. Overall, 99.5% of captions are unique. Caption lengths, measured in characters, have a global mean of 145.50, a standard deviation of 110.86, and a mode of 86. However, the mean and standard deviation are affected by a small number of extreme outliers, as caption lengths range from a minimum of 3 to a maximum of 5,330 characters. The typical caption is shorter: after excluding captions above the 95th percentile, the mean length drops to 128.62 characters and the standard deviation to 68.92. The distribution is therefore strongly right-skewed, with 99% containing at most 526 characters. Figure 3 shows the long-tailed distribution of caption lengths.

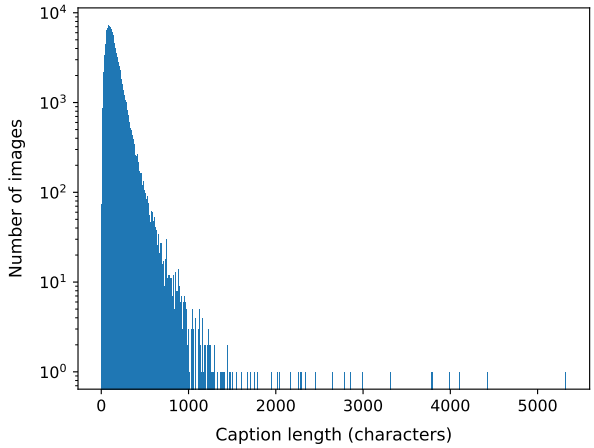


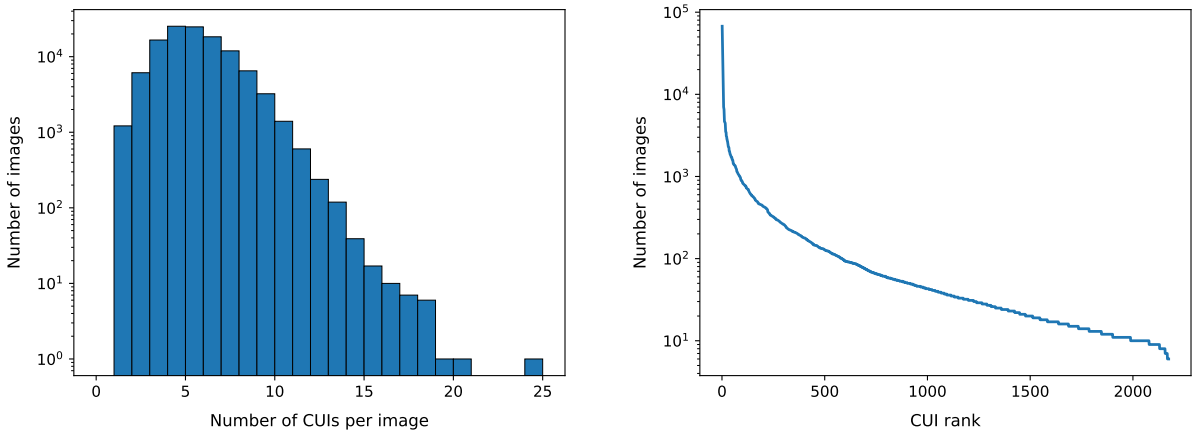
Figure 3: Distribution of caption length per image in the standard development data. The y-axis uses logarithmic scale to improve visibility of less frequent caption lengths.

2.2. Synthetic Data

The synthetic release contains 97,222 training, 19,239 validation, and 15,262 test images. For the synthetic variants of the tasks, ImageCLEF generated new captions for the images and then extracted new UMLS concept annotations from those generated captions. The organizers did not specify the exact method used to generate the synthetic captions. Therefore, we do not make further assumptions about the generation process.

2.2.1. Concept Detection Data Statistics

For the Synthetic Concept Detection development data, we repeated the same analysis. The split contains 2,174 unique CUIs. Each image is associated with an average of 5.05 concepts and a mode of 4 concepts per image, with the number of concepts ranging from 1 to 24. This is higher than in the standard split, where images have an average of 3.22 concepts and a mode of 2 concepts per image. Figure 4a shows the distribution of CUI counts per image.



(a) Distribution of CUI counts per image.

(b) Long-tailed CUI frequency distribution.

Figure 4: Synthetic Concept Detection label statistics. Both y-axes use logarithmic scale to improve visibility of less frequent values.

The synthetic concept-label distribution follows a similar long-tailed pattern, as shown in Figure 4b. A small set of frequent CUIs accounts for a large portion of the annotations, with the most common concepts appearing in nearly half of the development images. In contrast, the majority of CUIs occur in only a small percentage of images, with a global mean occurrence rate of 0.23% (≈271 images per CUI). Table 2 reports the most frequent CUIs and their occurrence rates. Some UMLS terms were unavailable in the mapping used for this analysis.

Table 2
Top 10 most frequent CUIs in the synthetic development data. Some UMLS Terms were unavailable.

Rank	Images	Occurrence%	CUI	UMLS Term
1	67,453	57.92%	C0205379	Unknown concept
2	48,450	41.60%	C0336721	Unknown concept
3	35,961	30.88%	C0817096	Chests
4	22,944	19.70%	C0000726	Abdomens
5	18,471	15.86%	C0037303	Bone structure of cranium
6	13,037	11.19%	C0030797	Pelvis
7	8,261	7.09%	C0221205	Unknown concept
8	7,010	6.02%	C0023216	Lower extremity
9	6,763	5.81%	C0242485	Unknown concept
10	6,530	5.61%	C0037949	Vertebral Column

According to the organizers, the Synthetic Concept Detection task does not provide the manually curated subset of CUIs used for the secondary evaluation of the standard Concept Detection task.

2.2.2. Caption Prediction Data Statistics

We repeated the caption-length analysis for the synthetic development split. Overall, 99.1% of captions are unique. Caption lengths, measured in characters, have a global mean of 191.84, a standard deviation

of 55.09, and a mode of 162, indicating that the synthetic captions are longer on average than the standard captions.

Although the synthetic caption-length distribution remains right-skewed, the effect of extreme outliers is much less pronounced than in the standard data. Caption lengths range from 53 to 702 characters, compared with a maximum of 5,330 in the standard split. This lower variability is also reflected in the standard deviation, which is 55.09 for the synthetic captions, almost half of the 110.86 observed for the standard captions. Figure 5 shows the caption-length distribution.

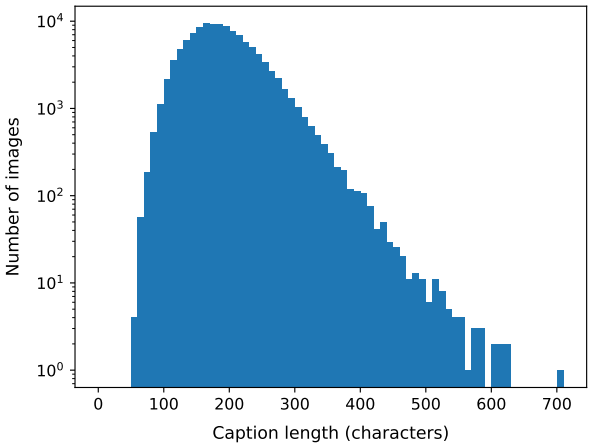


Figure 5: Distribution of caption length per image in the synthetic development data. The y-axis uses logarithmic scale to improve visibility of less frequent caption lengths.

Note. As per the organizers' instructions, the datasets were not mixed between the standard and synthetic tasks.

3. Methodology

In this section, we break down our approach for the Concept Detection and Caption Prediction tasks.

3.1. Concept Detection

We treat the Concept Detection subtask as a multi-label image classification problem. Given an input image I, the goal is to predict a set of UMLS CUIs associated with the image. Since multiple concepts may be assigned to the same image, each target is represented as a multi-hot vector $y \in \{0,1\}^C$, where C is the number of concepts in the vocabulary and $y_c = 1$ indicates that concept c is present.

Accordingly, each model outputs a vector of concept probabilities $\hat{y} \in [0,1]^C$. These probabilities are produced using independent sigmoid activations, since concept classes are not mutually exclusive and multiple CUIs may be valid for the same image. At inference time, the predicted CUI set is obtained by applying a decision threshold T to each concept probability, such that concept c is selected when:

$$\hat{y}_c \geq T$$

3.1.1. Data Preprocessing

Images were preprocessed using a shared pipeline across models. The ROCOv2 dataset contains radiology images with varying aspect ratios, and since every region of a medical image can contain clinically relevant information, we wanted to avoid force-fitting images to a square shape. However, we also wanted to avoid center-cropping images to the input size, which would risk removing important visual and medical information. To address this, we implemented the following resizing and padding

strategy: first, the longest side of each image was resized to the target input size, and the shorter side was then padded symmetrically to obtain a square image of the desired input size while preserving the original aspect ratio. Padding was performed using a fill value derived from the normalization mean of the corresponding pretrained model.

For models with available pretrained weights, we used the preprocessing parameters associated with those weights, like the interpolation mode, normalization mean, and standard deviation. When no pretrained-weight transform was available, we defaulted to ImageNet-style normalization with mean $(0.485, 0.456, 0.406)$ and standard deviation $(0.229, 0.224, 0.225)$.

For training images only, we also applied data augmentation to improve robustness. However, since the task involves medical images and content preservation is important, we kept the augmentations intentionally light. The augmentation pipeline included random affine transformations with small rotations of up to $4°$, translations of up to 2%, and scale changes between 0.95 and 1.05. These transformations were applied with probability 0.5. In addition, brightness and contrast jittering with magnitude 0.10 was applied with probability 0.3, and Gaussian blur with kernel size 3 and $\sigma \in [0.1, 1.0]$ was applied with probability 0.1. After augmentation, images were converted to floating-point tensors and normalized using the selected mean and standard deviation.

For validation and test inference, no random augmentation was applied. Images were processed deterministically using only aspect-ratio-preserving resizing, square padding, floating-point tensor conversion, and normalization.

3.1.2. Our System

Our Concept Detection system uses an ensemble of pretrained CNN and ViT backbones. We combine these backbone families to exploit both local convolutional and self-attention-based global features. All base models follow the same general architecture: a pretrained visual encoder that produces an image representation which is then passed to a task-specific multi-label classification head. Figure 6 provides an overview of the base model structure and final ensemble inference pipeline.

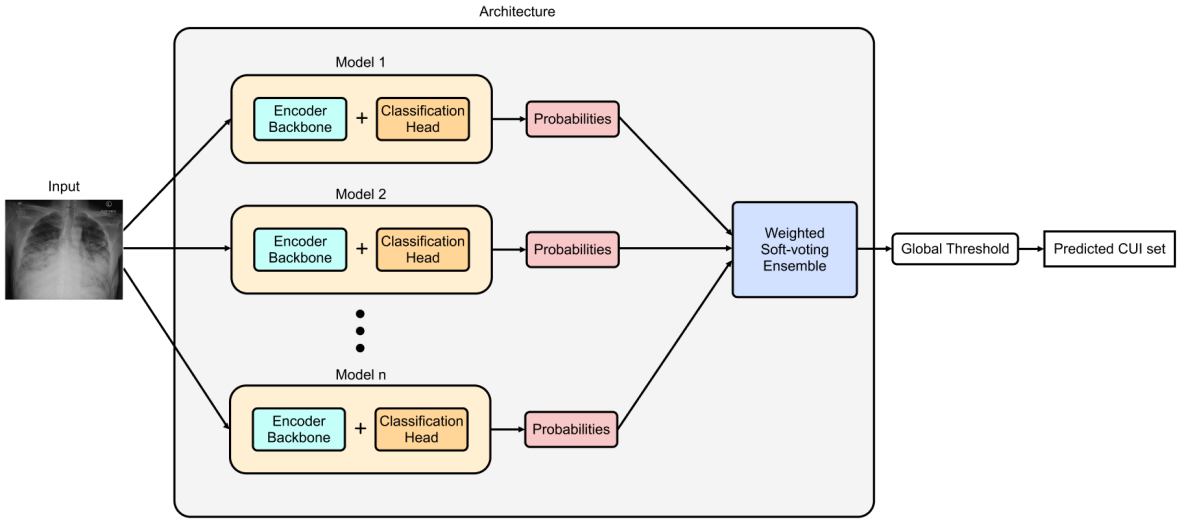


Figure 6: Concept Detection architecture diagram. Input image attribution: CC BY-NC Jackson et al. 2021.

Specifically, all encoder backbones were initialized with their default pretrained weights, using ImageNet-1K V1 or V2 depending on the model, and adapted to the task by replacing their original classifiers with dropout-regularized linear classification heads. In each case, the replacement head has the same two-layer structure: first, a dropout layer with a rate of 0.2 to reduce potential overfitting; second, a linear layer mapping the backbone's resulting feature representation to C output neurons, each corresponding to one CUI in the vocabulary. The resulting output vector contains one logit per

concept, allowing the same multi-label prediction setup to be used across all models. During training, the full model is fine-tuned end-to-end, with no frozen backbone layers.

We evaluated a set of candidate backbones spanning both convolutional and transformer-based architectures. The CNN-based candidates included ConvNeXt, ConvNeXt-Tiny [7], and EfficientNetV2-Small [8]. The Vision Transformer-based candidates included Swin-Small [9] and DINOv2 [10].

DINOv2 required a slightly different feature extraction strategy from the other backbones. We used the DINOv2 ViT-B/14 backbone loaded through the official PyTorch Hub interface. In Vision Transformer models, an image is split into patch tokens, and a special classification token, usually denoted as `[CLS]`, is added to the token sequence. After the transformer layers process the full sequence, the `[CLS]` token acts as a compact global representation of the image, while the patch tokens retain information associated with local image regions. The official DINOv2 implementation exposes both the normalized `[CLS]` token and the normalized patch-token representations through its `forward_features` function.

Instead of relying only on the `[CLS]` token, we constructed the final image representation by concatenating it with the mean-pooled patch-token representation:

$$h = [h_{\text{CLS}}; \bar{h}_{\text{patch}}].$$

This combines global image information from the classification token with aggregated patch-level information from the image tokens. The concatenated vector was then passed through a classification head consisting of layer normalization, dropout with rate 0.2, and a final linear layer with C output neurons. The additional layer normalization was used to stabilize the scale of the concatenated representation before the final classification layer. As with the other base models, the output contains one logit per CUI and follows the same multi-label prediction formulation.

3.1.3. Training Details

Loss Function. For the Concept Detection models, we used Asymmetric Loss (ASL), which is designed for multi-label classification with positive-negative imbalanced data [11]. This is particularly suitable for our case: as shown in the data analysis, each image in the standard development split is associated with only 3.22 CUIs on average out of a vocabulary of 2,646 concepts. Therefore, only a very small fraction of the target vector is positive for each image, while the vast majority of labels are negative. ASL addresses this imbalance by applying different focusing parameters to positive and negative labels and by reducing the contribution of easy negative examples.

Given a logit z_c for concept c, the predicted probability is computed using the sigmoid function, $p_c = \sigma(z_c)$. ASL defines separate terms for positive and negative labels. For positive labels, the loss term is

$$L_+ = (1 - p_c)^{\gamma_+} \log(p_c)$$

while for negative labels, ASL first applies asymmetric clipping with margin $m = 0.05$, using

$$p_m = \max(p_c - m, 0)$$

and then defines the negative term as

$$L_- = p_m^{\gamma_-} \log(1 - p_m)$$

This clipping suppresses very easy negative labels: when the predicted probability p_c is smaller than the margin m, the shifted probability p_m becomes zero and the negative term no longer contributes to the loss. This is useful in our setting because most CUIs are absent for each image.

The final per-concept loss is then

$$\mathcal{L}_c = -y_c L_+ - (1 - y_c) L_-$$

where $y_c \in \{0, 1\}$ indicates whether concept c is present in the ground truth. In our experiments, we used $\gamma_+ = 1$ and $\gamma_- = 4$, assigning stronger focusing to negative labels. This reduces the influence of

the abundant easy negatives in our sparse CUI target vectors, preventing them from dominating the optimization. The final loss is averaged over all concepts and batch samples.

We also experimented with more conventional alternatives, including per-class weighted binary cross-entropy with logits and even a linear combination of binary cross-entropy and ASL. However, pure ASL consistently achieved better validation performance and was therefore used as the final training objective for all Concept Detection models.

Optimizer. All Concept Detection models were optimized using AdamW [12], which decouples weight decay from the gradient-based parameter updates. This makes it suitable for our fine-tuning setting, where both pretrained backbone parameters and newly initialized classification head parameters are optimized together.

To account for the fact that the backbone already contains pretrained visual features, whereas the classification head must be learned from scratch, we used separate learning rates. The backbone was fine-tuned with a lower learning rate of 1×10^{-5}, while the newly initialized classification head used a higher learning rate of 1×10^{-4}.

Weight decay was set to 10^{-3} by default. For larger or more regularization-sensitive backbones, including ConvNeXt, ConvNeXt-Tiny, and DINOv2, weight decay was increased to 0.05. However, it was not applied to bias terms and one-dimensional parameters such as normalization weights and biases, positional embeddings, special tokens, or ConvNeXt layer-scale parameters. These parameters were excluded because they usually control offsets, normalization statistics, positional information, or scaling behavior rather than dense feature weights. Applying weight decay to them can unnecessarily constrain calibration.

The final optimizer configuration therefore used four parameter groups: backbone parameters with and without weight decay, and classification-head parameters with and without weight decay.

Learning-Rate Scheduler. We used a step-based learning-rate schedule consisting of a linear warmup followed by cosine annealing. The total number of scheduler steps is computed from the number of training epochs and the number of batches per epoch. During the warmup phase, the learning rate increases linearly from 1% of its target value to the full learning rate. The warmup duration is 5% of the total training epochs, with a forced minimum of one warmup epoch. After warmup, cosine annealing is applied for the remaining training steps, gradually decreasing the learning rate to a minimum value of 1×10^{-6}.

Training Setup. Each Concept Detection model was trained for 20–30 epochs with a batch size of 32 using random seed 42. Experiments were run on an NVIDIA GeForce RTX 3080 GPU with 10 GB of GDDR6X memory, using automatic mixed precision (AMP). During training, GPU memory usage was approximately 9.8 GB, depending on the backbone and input resolution. After each epoch, the model was evaluated on the validation split. We applied early stopping with a patience of 7 epochs, monitoring a recreated competition primary F1 score rather than validation loss. The best model checkpoint was selected according to this validation F1 score and restored when early stopping was triggered.

Final Ensemble. The final Concept Detection system used a weighted soft-voting ensemble for the best trained models. For each test image, every model produced a logit vector over the full CUI vocabulary; each logit is converted into a probability using independent sigmoid activations applied per CUI. These vectors of probabilities were then combined using model-specific weights:

$$\hat{y}_{ens} = \frac{\sum_{m=1}^{M} w_m \hat{y}^{(m)}}{\sum_{m=1}^{M} w_m},$$

where M is the number of models in the ensemble, $\hat{y}^{(m)}$ is the probability vector produced by model m, and w_m is its corresponding ensemble weight.

The final ensemble combined the best-performing validation checkpoints shown in Table 3. Model selection and ensemble weights were based on validation primary F1. ConvNeXt and Swin-Small were assigned the highest weights, as they achieved the best validation performance and brought architectural diversity to the ensemble.

Table 3
Final Concept Detection ensemble models and weights.

Model	Standard Ensemble Weights	Synthetic Ensemble Weights
ConvNeXt	0.5	0.4
Swin-Small	0.5	0.6
ConvNeXt-Tiny	0.1	–
DINOv2	0.1	–
EfficientNetV2	0.1	–

Inference. The final probability vector was converted into a predicted CUI set using a global decision threshold of $T = 0.6$. Thus, concept c was included in the final submission when its ensemble probability satisfied $\hat{y}_{\text{ens},c} \geq T$.

We tested different global thresholds on the validation set and found that more conservative predictions generally performed better. Thresholds with $T < 0.5$ substantially underperformed, while thresholds with $T > 0.5$ produced more selective predictions without being overly restrictive. Based on validation performance, $T = 0.6$ provided the best trade-off.

3.2. Caption Prediction

We treat the Caption Prediction subtask as an image-conditioned sequence generation problem. Given an input image I, the goal is to generate a caption S that describes the relevant visual and medical content of the image. Specifically, the model's output is an ordered sequence of natural-language tokens.

Let the target caption be represented as a token sequence $S = (s_1, s_2, \ldots, s_N)$, where N is the sequence length. The model estimates the conditional probability of the caption given the image as

$$P(S \mid I) = \prod_{t=1}^{N} P(s_t \mid s_{<t}, I),$$

where $s_{<t}$ denotes the tokens before position t. This autoregressive formulation means that each caption is generated one token at a time, with each token prediction conditioned on the image representation and the previously generated caption tokens. Then the token sequence is decoded into the final predicted caption.

3.2.1. Data Preprocessing

The Caption Prediction model uses two different visual encoders, BiomedCLIP [13] and Swin-Small [9], each of which requires different input preprocessing. Therefore, we did not re-use the single shared resize–pad–normalize pipeline from the Concept Detection task. Instead, each image was loaded once as an RGB image and then independently processed into two separate model-specific tensors.

However, during training, a light shared image-level augmentation was optionally applied before the encoder-specific preprocessing. In the final captioning setup, this consisted only of brightness and contrast jittering with magnitude 0.10, applied with probability 0.3. Stronger geometric augmentations, such as affine transformations and square padding, were also tested but not used in the final captioning configuration. Some of these stronger shared preprocessing steps were used in the synthetic variant experiments.

After the optional shared training transform, the same image was passed through two different preprocessing pipelines. The BiomedCLIP branch used the preprocessing function returned with the pretrained BiomedCLIP model, producing the `biomed_pixels` tensor with shape:

$$I_{\text{bio}} = [B, C_{\text{rgb}}, S, S]$$

where B is the batch size, $C_{\text{rgb}} = 3$ denotes the RGB channel dimension, and S is the target square input size.

The Swin-Small branch used the preprocessing transform created from the resolved pretrained Swin model configuration, producing the `swin_pixels` tensor with shape:

$$I_{\text{swin}} = [B, C_{\text{rgb}}, S, S]$$

These encoder-specific preprocessing functions perform the final image resizing, tensor conversion, and normalization required by each pretrained visual encoder.

For validation and test inference, no shared random augmentation was applied. The images were processed only through the deterministic encoder-specific preprocessing pipelines for BiomedCLIP and Swin-Small.

Caption text was also preprocessed before training. Each caption was cleaned and then tokenized using the tokenizer associated with `google/flan-t5-base`, the language model we used for caption generation. This tokenizer converts captions into subword token ID sequences compatible with FLAN-T5 [14, 15]. All caption sequences are padded or truncated to a maximum length of 80 tokens. After tokenization, all padding token IDs are replaced with -100, which is the ignore index used by `T5ForConditionalGeneration`, so that padding positions do not contribute to the built-in cross-entropy loss. Finally, the tokenizer output is returned as PyTorch tensors, and the resulting `input_ids` are used as the training labels.

3.2.2. Architecture

Our Caption Prediction model combines domain-specific visual encoding, spatial visual encoding, a trainable visual-language bridge, and a parameter-efficient language decoder. The architecture is built around four main components: a BiomedCLIP image encoder [13], a Swin-Small visual encoder [9], a Querying Transformer bridge, and a LoRA-adapted [16] FLAN-T5 [14, 15].

The overall captioning architecture follows the BLIP-2 design principle of connecting frozen visual encoders and a frozen language model through a lightweight trainable bridge [17]. Our use of complementary visual representations and lightweight adaptation was additionally motivated by design directions explored in the PCLmed 2024 ImageCLEFmedical Caption system [18]. Our choice to combine architecturally different encoders was also motivated by the strong performance of multi-model ensembling in our current Concept Detection Task solution.

BiomedCLIP is used as a biomedical visual-language encoder. Since it is pretrained on large-scale biomedical image-text pairs, it provides visual representations that are better aligned with biomedical concepts than general natural-image encoders [13]. In our architecture, BiomedCLIP is used to extract a global biomedical image representation. The BiomedCLIP encoder is initialized from its pretrained weights and kept frozen during training, so its role is to provide stable domain-aware visual features.

Swin-Small is used as a second visual encoder to provide spatial image features. Unlike the global BiomedCLIP embedding, Swin-Small produces a feature map that preserves spatial information from the image. This is useful for caption generation because radiology captions may depend on localized visual patterns and anatomical regions. The Swin-Small encoder is initialized with pretrained weights and used without its classification head. In the final configuration, it is also kept frozen during training.

The visual-language bridge connects the frozen visual encoders to the decoder through projection layers, normalization layers, and a Querying Transformer. Inspired by the Q-Former design used in BLIP-2, the Querying Transformer uses learned query tokens to attend to the fused image features and extract a compact visual memory for the language model. In our implementation, this bridge is fully

trainable and is responsible for learning how to transform frozen visual features into a representation that can be utilized by the language decoder.

FLAN-T5-base is used as the language-generation component of our model. It builds on the T5 text-to-text encoder-decoder architecture [14] and is further instruction fine-tuned [15], providing strong pretrained sequence-generation capability. In the final configuration, the original FLAN-T5 parameters are kept frozen and the model is adapted using LoRA.

LoRA adapts the frozen FLAN-T5 model by adding trainable low-rank updates to selected attention projection matrices while keeping the original weights frozen. In our implementation, LoRA was inserted into the FLAN-T5 attention projection layers q, k, v, and o, with rank $r = 8$, scaling factor $\alpha = 16$, dropout 0.05, and no trainable bias terms. Thus, only the low-rank adapter parameters were optimized, reducing the memory cost of training while keeping the main FLAN-T5 parameters fixed.

In the final training configuration, only the visual-language bridge and FLAN-T5 LoRA adapter parameters are trainable. This design allows the model to reuse strong pretrained visual and language representations while learning only the task-specific mapping needed for radiology caption generation.

The following paragraphs trace the flow of information through the model, showing how the visual features are encoded, fused, compressed, and passed to the decoder to produce the final caption. Figure 7 provides an overview of the complete Caption Prediction architecture.

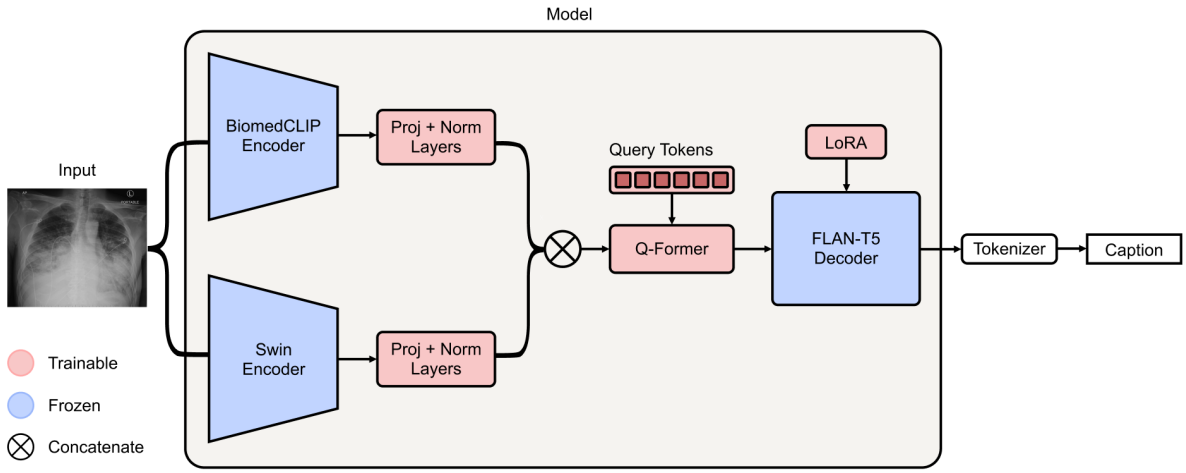


Figure 7: Caption Prediction model architecture diagram. Input image attribution: CC BY-NC Jackson et al. 2021.

Dual-Encoding Step. First, we pass the BiomedCLIP-preprocessed image tensor I_{bio} through the BiomedCLIP image encoder to obtain a global visual embedding for that image. This produces the tensor:

$$E = [B, C_{\text{bio}}]$$

where B is the batch size and C_{bio} is the BiomedCLIP feature dimension.

Using a linear layer, we project the BiomedCLIP embedding to the FLAN-T5 hidden dimension D producing:

$$E' = [B, D]$$

A layer-normalization module is then applied, and a singleton sequence dimension is added so that the BiomedCLIP representation finally becomes:

$$\tilde{E} = [B, 1, D]$$

This allows the BiomedCLIP embedding to be treated as a single visual token and makes it compatible for concatenation with the Swin token sequence.

Second, the Swin-preprocessed image tensor I_{swin} is passed through a Swin-Small encoder without its classification head. The encoder outputs a last-stage feature map with shape:

$$[B, C_{\text{swin}}, H_s, W_s]$$

To convert this feature map into a token sequence, we first permute the dimensions to $[B, H_s, W_s, C_{\text{swin}}]$, and then flatten the spatial dimensions to get tensor:

$$S = [B, N, C_{\text{swin}}]$$

where $N = H_s \times W_s$ is the number of spatial tokens and C_{swin} is the Swin feature dimension. These tokens are then projected to the FLAN-T5 hidden dimension D using a linear layer, followed again by layer normalization, producing the final Swin image tokens tensor:

$$S' = [B, N, D]$$

The normalization layers are used to reduce potential scale mismatch between the outputs of the two different visual encoders. After these steps both encoded token tensors have compatible hidden dimensions. We therefore concatenate the BiomedCLIP token tensor $[B, 1, D]$ and the Swin token tensor $[B, N, D]$ along the sequence dimension:

$$F = [\tilde{E}; S']$$

which produces the fused visual encoding:

$$F = [B, 1 + N, D]$$

This fused encoding is then passed to the Querying Transformer.

Q-Former Step. The fused visual encoding $F = [B, 1 + N, D]$ is fed to the Querying Transformer. This module is inspired by the Q-Former design in BLIP-2, where a small set of learned query tokens is used to extract a compact visual representation from image features. In our implementation, the Querying Transformer receives the fused image tokens as memory and uses Q learned query tokens as the target sequence.

First, the fused visual tokens are linearly projected to the Querying Transformer hidden dimension and used as the memory sequence:

$$M = \text{Linear}(F)$$

The model uses a fixed set of learned query tokens, which is expanded across the batch:

$$Q_0 = [B, Q, D]$$

where Q is the number of query tokens. Each query token can be interpreted as a trainable slot that learns what type of information to extract from the visual features. In our implementation, $Q = 32$.

Inside the Querying Transformer, the learned query tokens attend to the fused visual memory through multi-head cross-attention. In this operation, the query tokens act as the queries, while the fused visual tokens provide the keys and values. Conceptually, this follows scaled dot-product attention [19].

Thus, instead of passing all visual tokens directly to FLAN-T5, the model learns 32 query-based summaries of the image. Each learned query token aggregates information from both the BiomedCLIP global token and the Swin spatial tokens. After the Transformer decoder layers, the updated query-token sequence becomes the final visual memory:

$$V_{\text{mem}} = [B, Q, D]$$

This fixed-length visual memory tensor has the same hidden dimension D as FLAN-T5 and serves as the encoder-side representation for the decoder.

Decoding Step. Before decoding, the visual memory must be formatted so that it can be used as input to the FLAN-T5 decoder. Although we use FLAN-T5 mainly for its decoder, the full model follows an encoder-decoder architecture. However, since the image encodings are produced independently, we must bypass the default T5 text encoder step and pass the visual memory directly as encoder output.

To provide the visual memory in the expected format, we wrap it using the Hugging Face `BaseModelOutput` class. This wrapper does not modify the tensor or perform additional computation; it only exposes the visual memory as encoder output in the format expected by the T5 decoder.

We also create an attention mask for the visual memory. The mask has shape $B \times Q$, matching the batch and sequence dimensions of V_{mem}. In standard text inputs, this mask is used to distinguish valid tokens from padding tokens during attention computation. In our case, all entries in the mask are set to one, because the visual memory consists entirely of valid Q-Former output tokens and contains no padding positions. The FLAN-T5 decoder can then attend to these visual memory tokens through its cross-attention mechanism.

At this point, the decoder is used in two different modes. During training, it receives the visual memory together with the target caption tokens in order to compute next-token logits and loss. During inference, it receives only the visual memory and generates the caption autoregressively.

For the training path, V_{mem} serves as the encoder-side context during supervised training. With teacher forcing, the shifted target caption tokens are used as decoder input, and the decoder predicts a vocabulary-sized next-token logits tensor for each caption position:

$$Z_{\text{logits}} = [B, L, N_{\text{vocab}}]$$

where B is the batch size, L is the target caption length, and N_{vocab} is the FLAN-T5 vocabulary size. These logits are then used to compute the sequence-to-sequence language-modeling loss.

For the generation path, the decoder produces the caption autoregressively, starting from its initial decoder token and producing one token at a time. At each step, the next-token distribution is conditioned on V_{mem} and on the caption tokens generated so far. The output is a token sequence that is decoded into natural language using the model's tokenizer.

3.2.3. Training Details

Loss Function. The Caption Prediction model was trained using FLAN-T5's built-in `T5ForConditionalGeneration` language-modeling loss, corresponding to token-level cross-entropy over the target caption sequence. The tokenized caption is provided as the label sequence, and the model computes the loss over the predicted decoder logits.

Let the target caption be $S = (s_1, s_2, \ldots, s_N)$, where each s_t is a token ID from the FLAN-T5 vocabulary. At each decoding step t, the model predicts a distribution over the vocabulary conditioned on the visual memory and the previous target tokens:

$$p_\theta(s_t \mid s_{<t}, V) = \text{softmax}(z_t)_{s_t}$$

where z_t denotes the decoder logits at position t. The training objective is the negative log-likelihood of the target caption tokens:

$$\mathcal{L} = -\frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} \log p_\theta(s_t \mid s_{<t}, V),$$

where $\mathcal{M}$ is the set of non-padding target positions. The loss is computed from the FLAN-T5 output during the forward pass and is used to optimize the trainable visual-language bridge and LoRA adapter parameters.

Optimizer. The Caption Prediction model was optimized using AdamW [12]. In the final configuration, the BiomedCLIP and Swin-Small encoders were kept frozen, and the FLAN-T5 backbone was

adapted only through LoRA. Therefore, the optimizer was applied only to the trainable visual-language bridge parameters and the FLAN-T5 LoRA adapter parameters.

We used separate learning rates for these trainable components. The bridge parameters, mainly the Q-Former along with the visual projection layers and normalization layers, were trained with a learning rate of 3×10^{-5}. The LoRA adapter parameters in FLAN-T5 were trained with a learning rate of 2×10^{-5}. The bridge used a slightly higher learning rate because it had to learn the task-specific mapping from frozen visual features to the FLAN-T5 hidden space, whereas the LoRA adapters used a lower learning rate to adapt the pretrained language model more conservatively.

Weight decay was set to 0.01 for regularized bridge parameters. It was not applied to bias terms, normalization parameters, learned query tokens, positional or embedding-related parameters, or LoRA parameters. These parameters were excluded because they mainly control offsets, scaling behavior, learned token representations, or adapter updates rather than dense feature transformations. The final optimizer configuration therefore used separate decay and no-decay groups for the bridge parameters, together with separate LoRA parameter groups optimized without weight decay.

Learning-Rate Scheduler. We used a step-based learning-rate schedule consisting of a linear warmup followed by cosine decay. The total number of scheduler steps was computed from the number of training epochs and the number of optimizer steps per epoch, accounting for gradient accumulation. During the warmup phase, the learning rate increased linearly from 1% of its target value to the full learning rate. The warmup duration was set to 5% of the total optimizer steps, with at least one warmup step. After warmup, a cosine decay schedule was applied for the remaining training steps. Rather than decaying to a fixed absolute minimum learning rate, the scheduler independently decayed each parameter group's learning rate to 20% of its respective full learning rate. This choice was motivated by underfitting in early experiments, where validation loss continued to decrease steadily across epochs.

Training Setup. Each Caption Prediction model was trained for 26–32 epochs with a batch size of 24 using random seed 42. Experiments were run on an NVIDIA GeForce RTX 3080 GPU with 10 GB of GDDR6X memory, using automatic mixed precision (AMP). During training, GPU memory usage was approximately 9.8 GB. After each epoch, the model was evaluated on the validation split. We applied early stopping with a patience of 3 epochs, monitoring validation loss. The best model checkpoint was selected accordingly and restored when early stopping was triggered.

Inference. Caption generation was performed using beam search. Beam search maintains several high-scoring partial captions at each step and selects the best complete sequence at the end. For the final submissions, we used a beam width of 5 and limited generation to 64 new tokens. We tested different no-repeat n-gram values and found that $n = 4$ obtained the highest official test score. Smaller values were more restrictive for repeated phrases, which may still be valid in medical captions, while disabling the constraint led to a slightly lower overall score. The length penalty was set to 1.0, so no additional preference for shorter or longer captions was introduced. The generated token IDs were decoded with the FLAN-T5 tokenizer, skipping special tokens, and the resulting text was normalized into a single-line caption for the official submission file.

4. Results

This section presents the official test metrics and rankings for our final submissions to the standard and synthetic Concept Detection and Caption Prediction subtasks for ImageCLEFmedical Caption 2026.

4.1. Concept Detection Results

Table 4 reports the standard validation performance of the individual Concept Detection models and the final weighted ensemble. The weighted ensemble improved over each individual model on the standard

Table 4
Standard Concept Detection validation performance of individual models and the final ensemble.

Model	Primary F1	Secondary F1
ConvNeXt	0.58311	0.9168
Swin-Small	0.58635	0.9209
ConvNeXt-Tiny	0.58317	0.9177
DINOv2	0.58173	0.9154
EfficientNetV2	0.58387	0.9173
Weighted ensemble	0.59014	0.9253

Table 5
Official test metrics and rankings for the Concept Detection subtasks.

Concept Detection			
Task	Primary F1	Secondary F1	Ranking
Standard	0.5721	0.9503	5th
Synthetic	0.5609	–	1st

validation set for both the primary and secondary F1 scores.

Table 5 shows the official Concept Detection test results, with our submissions ranking 5th on the standard subtask and 1st on the synthetic subtask. Our weighted ensemble produced the winning submission for the Synthetic Concept Detection subtask, while also remaining competitive on the standard variant. The primary F1 scores of the two submissions are relatively close, suggesting similar absolute performance across the two annotation settings, even though the relative ranking was higher on the synthetic task. For the standard subtask, the high secondary F1 indicates better performance on the manually curated concept subset, which mainly covers image modality and anatomical body region. This suggests that the model captured these broad visual distinctions well. Since the synthetic subtask does not include a secondary score, this metric cannot be compared across variants.

4.2. Caption Prediction Results

Table 6 shows the official Caption Prediction test results, with our submissions ranking 4th on the standard subtask and 2nd on the synthetic subtask.

Table 6
Official test metrics and rankings for the Caption Prediction subtasks.

Caption Prediction		
Metric	Standard	Synthetic
Overall	0.3516	0.5136
Relevance	0.5288	0.6542
Factuality	0.1743	0.3731
BERTScore	0.6130	0.7195
ROUGE-1	0.2728	0.5555
Similarity	0.9113	0.8902
BLEURT	0.3183	0.4515
MedCAT	0.1812	0.4677
AlignScore	0.1674	0.2784
Ranking	4th	2nd

During Caption Prediction model development, validation loss was used to monitor training, but we did not compute caption-level metrics such as ROUGE, BERTScore, BLEURT, and AlignScore for

decoding variants during validation. Since decoding configurations only affect the generated captions at inference time, they are not directly reflected in the teacher-forced validation loss. Therefore, different decoding settings were compared using the test metrics from the available submitted runs.

To make the Caption Prediction decoding configuration more traceable, Table 7 reports the official test overall scores for submitted Synthetic Caption Prediction runs that used the same trained caption model but different no-repeat n-gram settings. All runs used beam search with beam width 5, a maximum of 64 generated tokens, and length penalty 1.0. These results provide run-level evidence for the no-repeat n-gram setting used in the final submitted configuration.

Table 7
Official Caption Prediction test performance for submitted no-repeat n-gram decoding variants using the same caption model.

No-repeat n-gram	Overall score
None	0.5081
3	0.5083
4	**0.5136**

Among these submitted runs, the configuration with no-repeat n-gram set to 4 obtained the highest official test overall score and corresponds to the final submitted configuration.

The Caption Prediction results show a clear pattern: Our model performed noticeably better on the synthetic variant across most metrics. The largest differences appear in ROUGE, MedCAT, and the overall factuality score, suggesting that the synthetic annotations may be more regular or internally consistent. However, since the exact generation process for the synthetic captions and concepts is not specified by the organizers, we cannot make a definitive claim about the cause of this difference.

Although our standard Caption Prediction submission ranked 4th overall, AlignScore was its most distinctive metric, achieving the highest value among all submissions. This suggests that our architecture generated captions that were comparatively well aligned with the reference captions in factual and semantic aspects measured by AlignScore.

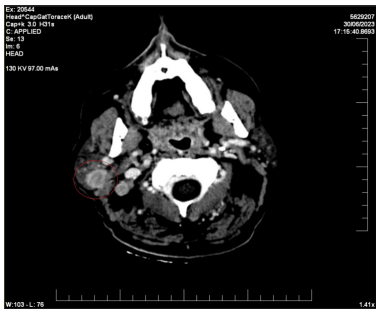
At the same time, the other metrics were competitive, but did not stand out to the same extent as AlignScore. This indicates that the alignment measured by AlignScore did not always translate into high lexical overlap, exact concept extraction, or reference-style phrasing. In other words, the model produced captions that are comparatively well aligned in meaning, but not always detailed or phrased in the same way as the ground-truth captions.

Figure 8 presents two qualitative examples from the standard Caption Prediction validation set to support the interpretation of the metric results. The examples show cases where the generated caption remains semantically aligned with the ground-truth caption, while differing in phrasing or in the level of detail, as illustrated in Figures 8a and 8b.

5. Conclusion

We presented our submissions to both the standard and synthetic Concept Detection and Caption Prediction subtasks of ImageCLEFmedical Caption 2026. The Concept Detection system was based on a weighted ensemble of pretrained CNN and Vision Transformer backbones, while the Caption Prediction system used a dual-visual-encoder, single-decoder architecture with a trainable Querying Transformer bridge.

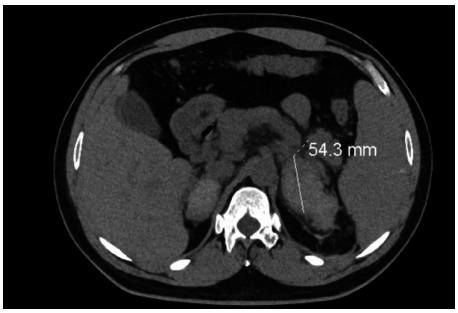
For Concept Detection, the results indicate that weighted soft-voting was a viable strategy for the sparse multi-label CUI prediction setting. The ensemble remained competitive on the standard task and achieved its strongest relative result on the synthetic variant. This suggests that combining architecturally diverse visual backbones helped produce robust concept predictions across both annotation settings.



ID: ImageCLEFmedical_Caption_2026_valid_13791

Ground-truth Caption
CECT showing a nodular mass within the right parotid gland (red circle). CECT: contrast-enhanced computed tomography.
Generated Caption
Contrast-enhanced computed tomography (CT) scan of the neck showing a mass in the right parotid gland.

(a) Semantic match with different wording.



ID: ImageCLEFmedical_Caption_2026_valid_5496

Ground-truth Caption
A transverse view of a non-contrast CT scan of the abdomen and pelvis shows a large left-sided adrenal hemorrhage.
Generated Caption
Computed tomography (CT) scan of the abdomen and pelvis showing a left adrenal hemorrhage.

(b) Correct main finding with less detail.

Figure 8: Qualitative examples from the standard Caption Prediction validation set. Each subfigure includes an input image, its corresponding ground-truth caption, and the caption generated by the submitted model.

For Caption Prediction, the results suggest that the dual-visual-encoder, single-decoder architecture with a Q-Former bridge produced competitive results for image-conditioned caption generation, especially on the synthetic variant. The AlignScore result on the standard task further suggests that the model's generated captions retained semantic and factual alignment, even when lexical overlap and explicit concept matching were not equally distinctive.

For future work, there are several extensions we would have liked to incorporate into our models but could not fully implement in time for the final submission. For Concept Detection, we would like to explore modality-aware modeling, where modality-related CUIs are first used to route images to separate models trained for specific imaging modalities, such as X-ray, CT, MRI, ultrasonography and angiography. This could help improve performance on sparse or underrepresented modalities that may not be handled optimally with a single shared classifier. We would also like to investigate per-label threshold optimization instead of using one global decision threshold for all CUIs.

For Caption Prediction, future work could explicitly condition the decoder on the predicted CUI set, so that concept-level predictions can guide caption generation. Caption reranking could also be explored to select more factual and concept-consistent outputs. Finally, if computational resources allowed, we would like to test different visual encoder models and configurations in which some encoder layers are fully fine-tuned rather than kept frozen.

Acknowledgments

We would like to thank the Athens University of Economics and Business BALab for providing the computational resources used for the implementation and experiments reported in this work.

Reproducibility and Code Availability

This work was implemented in Python using PyTorch. The source code is publicly available at: https://github.com/GeorgeKarandreas/image-clef-med-caption-task-2026.

Declaration on Generative AI

During the preparation of this work, the first author used ChatGPT in order to: Grammar and spelling check, Paraphrase and reword. After using this tool, the authors reviewed and edited all assisted text as needed and take full responsibility for the final content of the paper.

References

[1] B. Ionescu, H. Müller, D. Stanciu, A. Radu, R. Bolborici, M. Negru, A. Ene, V. Vasilescu, A.-A. Nicolae, L. Ştefan, M. Constantin, M. Dogariu, A. Andrei, H. Damm, T. M. G. Pakull, A. Ben Abacha, A. García Seco de Herrera, C. M. Friedrich, R. Brüngel, L. Reinartz, H. Schäfer, C. S. Schmidt, B. Bracke, P. Nath, B. Eryılmaz, M. Hjuler, D. Fabre, C. Lemaire, B. Lecouteux, D. Schwab, D. Dimitrov, M. S. Hee, M. Ahsan, S. Ahmad, D. Zlatkova, G. Pachov, Z. Xie, P. Nakov, I. Koychev, J. E. Heras Rivera, D. K. Low, W. Yim, J. Ruzevick, D. Child, M. Kurt, Z. Sun, F. Xia, M. Yetisgen, A. Radzhabov, Y. Prokopchuk, V. Kovalev, D. Karpenka, S. A. Hicks, S. Gautam, M. A. Riegler, V. Thambawita, P. Halvorsen, M. El Sakka, J. Mothe, A. Băicoianu, C. Florea, M. Ivanovici, Overview of ImageCLEF 2026: Multimodal challenges in medicine, science, agritech, and security, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Springer Lecture Notes in Computer Science LNCS, Jena, Germany, 2026.

[2] H. Damm, T. M. G. Pakull, L. Reinartz, B. Bracke, B. Eryılmaz, P. Nath, R. Brüngel, C. S. Schmidt, H. Schäfer, A. Ben Abacha, A. García Seco de Herrera, H. Müller, C. M. Friedrich, Overview of ImageCLEFmedical 2026—medical concept detection and caption generation with synthetic data extensions, in: CLEF2026 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, Jena, Germany, 2026.

[3] O. Bodenreider, The Unified Medical Language System (UMLS): integrating biomedical terminology, Nucleic Acids Research 32 (2004) D267–D270. doi:10.1093/nar/gkh061.

[4] J. Rückert, L. Bloch, R. Brüngel, A. Idrissi-Yaghir, H. Schäfer, C. S. Schmidt, S. Koitka, O. Pelka, A. B. Abacha, A. G. S. de Herrera, H. Müller, P. Horn, F. Nensa, C. M. Friedrich, ROCOv2: Radiology objects in COntext version 2, an updated multimodal image dataset, Scientific Data 11 (2024). doi:10.1038/s41597-024-03496-6.

[5] O. Pelka, S. Koitka, J. Rückert, F. Nensa, C. M. Friedrich, Radiology Objects in COntext (ROCO): A Multimodal Image Dataset, in: D. Stoyanov, Z. Taylor, S. Balocco, R. Sznitman, A. Martel, L. Maier-Hein, L. Duong, G. Zahnd, S. Demirci, S. Albarqouni, S.-L. Lee, S. Moriconi, V. Cheplygina, D. Mateus, E. Trucco, E. Granger, P. Jannin (Eds.), 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, Springer International Publishing, Cham, 2018, pp. 180–189.

[6] A. Chatzipapadopoulou, I. Pantelidis, F. Charalampakos, M. Samprovalaki, G. Moschovis, P. Kaliosis, K. V. Dalakleidi, J. Pavlopoulos, I. Androutsopoulos, AUEB NLP group at ImageCLEFmedical Caption 2025, in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, Madrid, Spain, 2025. URL: https://ceur-ws.org/Vol-4038/paper_186.pdf.

[7] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A ConvNet for the 2020s, in: 2022

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 11976–11986. doi:10.1109/CVPR52688.2022.01167.

[8] M. Tan, Q. Le, EfficientNetV2: Smaller Models and Faster Training, in: Proceedings of the 38th International Conference on Machine Learning (ICML 2021), PMLR, 2021, pp. 10096–10106.

[9] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin Transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2021), 2021, pp. 10012–10022.

[10] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jégou, J. Mairal, P. Labatut, A. Joulin, P. Bojanowski, DINOv2: Learning robust visual features without supervision, Transactions on Machine Learning Research (2024). URL: https://openreview.net/forum?id=a68SUt6zFt, Featured Certification.

[11] T. Ridnik, E. Ben-Baruch, N. Zamir, A. Noy, I. Friedman, M. Protter, L. Zelnik-Manor, Asymmetric Loss for Multi-Label Classification, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE Computer Society, Los Alamitos, CA, USA, 2021, pp. 82–91. URL: https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00015. doi:10.1109/ICCV48922.2021.00015.

[12] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: International Conference on Learning Representations (ICLR 2019), 2019.

[13] S. Zhang, Y. Xu, N. Usuyama, H. Xu, J. Bagga, R. Tinn, S. Preston, R. Rao, M. Wei, N. Valluri, C. Wong, A. Tupini, Y. Wang, M. Mazzola, S. Shukla, L. Liden, J. Gao, A. Crabtree, B. Piening, C. Bifulco, M. P. Lungren, T. Naumann, S. Wang, H. Poon, A multimodal biomedical foundation model trained from fifteen million image–text pairs, NEJM AI 2 (2024). URL: https://ai.nejm.org/doi/full/10.1056/AIoa2400640. doi:10.1056/AIoa2400640.

[14] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (2020) 1–67.

[15] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, et al., Scaling instruction-finetuned language models, Journal of Machine Learning Research 25 (2024) 1–53.

[16] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al., LoRA: Low-Rank Adaptation of Large Language Models, in: International Conference on Learning Representations (ICLR 2022), 2022.

[17] J. Li, D. Li, S. Savarese, S. Hoi, BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models, in: Proceedings of the 40th International Conference on Machine Learning (ICML 2023), PMLR, 2023, pp. 19730–19742.

[18] B. Yang, Y. Yu, Y. Zou, T. Zhang, PCLmed: Champion Solution for ImageCLEFmedical 2024 Caption Prediction Challenge via Medical Vision-Language Foundation Models, in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024), volume 3740 of *CEUR Workshop Proceedings*, CEUR-WS.org, Grenoble, France, 2024, pp. 1763–1774. URL: https://ceur-ws.org/Vol-3740/paper-164.pdf.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is All you Need, in: Advances in Neural Information Processing Systems 30: 31st Conference on Neural Information Processing Systems (NIPS 2017), 2017.