REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

DEEP MULTIMODAL LEARNING WITH VISION-AND-LANGUAGE TRANSFORMERS

17011086 — Mesut Şafak BILICI
17011041 — Enes Sadi UYSAL

SENIOR PROJECT

Advisor
Assoc. Prof. Dr. Mehmet Fatih AMASYALI

June, 2022

© All rights reserved to Yıldız Technical University, Computer Engineering Department.

ACKNOWLEDGEMENTS

Science is the co-production of the humanity, thus, it should be as free as breathing. We would like to express our appreciation and gratitude to Assoc. Prof. Dr. Mehmet Fatih Amasyali as our professor and advisor, and to those who contributed to and liberated science without any expectations.

Mesut Şafak BILICI
Enes Sadi UYSAL

TABLE OF CONTENTS

LIST OF ABBREVIATIONS v

LIST OF FIGURES vi

LIST OF TABLES viii

ABSTRACT ix

ÖZET x

1 INTRODUCTION 1

2 LITERATURE REVIEW 3

3 FEASIBILITY 5
- 3.1 Technical Feasibility 5
  - 3.1.1 Software Feasibility 5
  - 3.1.2 Hardware Feasibility 5
- 3.2 Legal Feasibility 6
- 3.3 Economic Feasibility 6
- 3.4 Workforce and Time Feasibility 7

4 SYSTEM ANALYSIS 8
- 4.1 Use Case Diagram 9

5 BACKGROUND 10
- 5.1 Transformers 10
  - 5.1.1 Self-Attention 11
- 5.2 Pre-Trained Language Models 13
  - 5.2.1 BERT 13
- 5.3 Vision Transformer 15

6 SYSTEM DESIGN: DEEP MULTIMODAL LEARNING WITH VISION-AND-LANGUAGE TRANSFORMERS 17
- 6.1 Retrieval and Ranking 18

6.1.1 Image-Text Matching 19
6.1.2 Deep Metric Learning for Retrieval 22
6.2 Visual Question Answering 24
6.2.1 Question Answering in Natural Language Processing 25
6.2.2 Visual Question Answering (VQA) 25
6.2.3 Benchmark Datasets in VQA Task 26
6.2.4 Metrics and Loss Functions in VQA Task 27

7 RESULTS AND PERFORMANCE ANALYSIS 29
7.1 Image-Text Matching Model 29
7.2 Metric Learning Model 31
7.3 VQA 35
7.4 DAQUAR
(DAtaset for QUestion Answering on Real-world images) 35

8 TIR: A NEW BENCHMARK FOR "T"URKISH "I"MAGE "R"ETRIEVAL AT SCALE 39
8.1 Dataset Characteristics 40
8.2 Results 42

9 APPLICATIONS 47
9.1 A Vector Search Engine for Images 47
9.2 Multilingual Visual Question Answering Tool 48

References 50

Curriculum Vitae 54

LIST OF ABBREVIATIONS

NLP	Natural Language Processing
CBOW	Continuous Bag of Words Model
L1	First Language
UG	Universal Grammar
VQA	Visual Question Answering
IR	Information Retrieval
RNN	Recurrent Neural Network
seq2seq	Sequence to Sequence
MLM	Masked Language Modeling
NSP	Next Sentence Prediction
RoI	Region of Interest
OOV	out-of-vocabulary
docid	Document ID

LIST OF FIGURES

Figure 3.1 Gantt Diagram 7
Figure 4.1 The Use Case Diagram 9
Figure 5.1 The Transformer. 11
Figure 5.2 Self-attention block 12
Figure 5.3 BERT Model 14
Figure 5.4 Fine-Tuning Modification of BERT for Sequence Classification . 15
Figure 5.5 Vision Transformer Model 16
Figure 6.1 Document Retrieval 18
Figure 6.2 Image Retrieval 19
Figure 6.3 Attention Matrices in DistilBERT and ViT. Attention layer and heads are choosen randomly. 20
Figure 6.4 Multimodal Transformer Architecture 21
Figure 6.5 Modeling Deep Metric Learning for Retrieval 23
Figure 6.6 QA System 25
Figure 6.7 Generative Approach 26
Figure 6.8 Answer Space Approach 26
Figure 6.9 Sampled Examples and Distribution of Questions in VQA Dataset 27
Figure 6.10 Sampled Examples and Distribution of Questions in DAQUAR Dataset 28
Figure 7.1 Twelve retrieved images for each four query. Images are stored and ranked from 4000 images. 30
Figure 7.2 Twelve retrieved images for each four query. Images are stored and ranked from 4000 images. 31
Figure 7.3 Twelve retrieved images for each four query. Images are stored and ranked from 7000 images. 33
Figure 7.4 Twelve retrieved images for each four query. Images are stored and ranked from 7000 images. 34
Figure 7.5 Model's Prediction Examples for VQA 37
Figure 7.6 Model's Prediction Examples for DAQUAR 38
Figure 8.1 LAION Dataset 39
Figure 8.2 Most frequent 70 words from TIR dataset 40

Figure 8.3 TIR Image Retrieval Examples (1): Twelve retrieved images for each four query. Images are stored and ranked from 150000 images. 43
Figure 8.4 TIR Image Retrieval Examples (2): Twelve retrieved images for each four query. Images are stored and ranked from 150000 images. 44
Figure 8.5 TIR Image Retrieval Examples (3): Twelve retrieved images for each four query. Images are stored and ranked from 150000 images. 45
Figure 8.6 TIR Image Retrieval Examples (4): Twelve retrieved images for each four query. Images are stored and ranked from 150000 images. 46
Figure 9.1 Demo Website for Retrieval Model 47
Figure 9.2 Demo Website for VQA Model 48
Figure 9.3 Turkish Output 49
Figure 9.4 English Output 49

LIST OF TABLES

Table 3.1 Python Packages and Version Dependencies 6
Table 3.2 Licenses of Packages 6
Table 7.1 Test Set Metrics for Flickr30k on Image-Text Matching Objective 30
Table 7.2 Test Set Metrics for Flickr30k on Deep Metric Learning Objective 32
Table 7.3 Different experimental results in percentage on VQA Dataset. @k indicates top-k accuracy. 35
Table 7.4 Different experimental results in percentage on DAQUAR Dataset. @k indicates top-k accuracy. 36
Table 8.1 Query Translations/Literal Translations for TIR Image Retrieval Examples 41
Table 8.2 Image Resolution Statistics of TIR. 41

ABSTRACT

DEEP MULTIMODAL LEARNING WITH VISION-AND-LANGUAGE TRANSFORMERS

Mesut Şafak BILICI
Enes Sadi UYSAL

Department of Computer Engineering
Senior Project

Advisor: Assoc. Prof. Dr. Mehmet Fatih AMASYALI

In recent years, Transformer model has gained high impact in Natural Language Processing. With Transformer based pre-trained language models, it achieved state-of-the-art performance results for downstream tasks such as understanding, comprehension and generation. As a result, Transformer based pre-trained image models are appreciated. In the past year, multimodal Transformers are studied for representing image and text pairs at once. This allows us to develop applications like image search engines, visual question answering systems.

In this study, it is aimed to research details of Vision-and-Language Transformers on different datasets with different objectives. We use such model designs on two downstream tasks: Image Retrieval and Visual Question Answering (VQA). These models are trained on Flickr30k, VQA and DAQUAR datasets; which are the common benchmarks in the literature. We discuss performances and outputs of different models and interrogate which one should be chosen for which task.

On account of latest developments on Vision-and-Language Transformers for English, we collect and compile a new dataset, which is called TIR (**T**urkish **I**mage **R**etrieval), and which can be considered as first in the image retrieval domain. With TIR dataset, a minimalist CLIP model is trained for image retrieval.

Keywords: Image Retrieval, Visual Question Answering, Metric Learning, Multimodal Learning, Transformers

ÖZET

GÖRÜNTÜ VE DİL TRANSFORMATÖRÜ İLE DERİN MULTIMODAL ÖĞRENME

Mesut Şafak BILICI
Enes Sadi UYSAL

Bilgisayar Mühendisliği Bölümü
Bitirme Projesi

Danışman: Doc. Dr. Mehmet Fatih AMASYALI

Son yıllarda Transformatör modelleri Doğal Dil İşleme alanında önemli bir yer kazandı. Önceden eğitilmiş Transformatör temelli dil modelleri; anlama, kavrama ve üretme görevlerinde en başarılı skorlara ulaşmıştır. Bunun sonucunda Transformatör temelli önceden eğitilmiş resim modelleri de ortaya çıkmıştır. Geçen sene içinde, resim ve metin Transformatör modelleri ile birlikte Görsel-Metin çiftleri tek seferde ifade edilmiştir. Bu sayede, görsel arama motorları, görsel soru cevaplama sistemleri gibi uygulamalar geliştirebiliriz.

Bu çalışmada, Görsel-Metin Transformatörlerinin detaylarının farklı veri setleri üzerinde farklı amaçlarla araştırılması amaçlanmıştır. Bu tür model tasarımlarını iki alt görevde kullanıyoruz: Görüntü Alma ve Görsel Soru Yanıtlama (GSY). Bu modeller Flickr30k, VQA ve DAQUAR veri kümeleri üzerinde eğitilmiştir; bunlar literatürdeki sık rastlanan veri setleridir. Farklı modellerin performanslarını ve çıktılarını tartıştık ve hangisinin hangi görev için seçilmesi gerektiğini sorguladık.

Bu çalışmada ayrıca, Görsel-Metin Transformatörlerdeki son gelişmeleri baz alarak, literatüre yeni Türkçe Görsel-Metin veriseti kazandırdık. Aynı zamanda bu veri seti ile görüntü alımı için minimalist bir CLIP modeli eğitildi.

Anahtar Kelimeler: Görüntü Çıkarımı, Görsel Soru Cevaplama, Metrik Öğrenme, Multimodal Öğrenme, Transformatör

1
INTRODUCTION

First language (L1) learning starts from the birth of an infant. This language learning course is not determined by a pre-defined prescription, as stated by Noam Chomsky [1]. It is the biological genes from our ancestors, which are the first priors to language learning. Traditional biolingustics have only concerned on minimalist program, principles & parameters, and Universal Grammar (UG), when it comes to learning a language. However, language learning should be parameterized by embodiment, as well. Modern language acquisiton theories include not only syntax, but interactions between the infant and the environmental ingredients. These ingredients can be classified as audio signals, visual inputs, causal reasonings [2] which are captured by sensorimotor systems (includes ears, eyes etc.) of learner. We sometimes call interaction of these environmental ingredients as "multimodal" interactions. It is unexceptionable that when the learning course is supported by the visual supervision, it will be more quicker and easier to learn. As an example, a classical children novel, entitled as Frindle, which is about a child who renames object pencil as "frindle". It is impossible to explain frindle to someone else who do not know what is frindle, without any visual supervision (or without any distributional semantics). If you tell someone "I forgot my frindle today" while showing a pencil picture, this will be the easiest and the painless way to teach what is frindle.

This motivation gives us many practical and theoretical aspects, when our mission is to teach machines to learn and infer. The most popular and state-of-the art way to teach machines is done by Neural Networks. Over the last five decades, they are modernized and developed by scientists by considering different modalities like text [3], image [4], signal/audio [5], physiological signals [6] etc. and tasks like Image Classification [7], Object Detection [8], Natural Language Understanding [9], Forecasting [10] etc.

Taking into account all of these, our main concern is to combine those modalities and align them, to learn a better representative space, which is supported by different modalities rather than one. Especially, our aim is to learn visual representations with natural language supervision.

This thesis presents latest developments on multimodal "image-and-text" learning. Traditional approaches for this problem generally include a Word Embedding model (for example CBOW) for text representation, and a grid structured model (for example Convolutional Neural Network) for image representation. By following latest developments on Natural Language Processing (NLP) and Computer Vision fields, we build our models and conduct our experiments on only Transformer model [11]. These terms are explained in detail at Related Work and Background sections.

Image-and-Text Transformers can be used for diverse practical tasks such as ranking: image retrieval, text retrieval, text-image matching; Visual Question Answering (VQA), and Zero Shot Image Classification. We approach these tasks in non-generative way, due to its stability and computational problems. We thus learn to align image and text pair, with different objectives.

At last, the main contribution is we present a novel image-text pair benchmark, called TIR (**T**urkish **I**mage **R**etrieval), for image retrieval. This benchmark is collected from many sources, which are grouped in a common and free database LAION-5B. After collecting and compiling the TIR benchmark, we train a minimalist implementation of CLIP model [12] and create a image search engine based on semantic vectors.

In this thesis; Section 2 introduces priors to our work and literature review, Section 3 feasibility which includes software and hardware dependencies for reproducibility, Section 4 introduces system analysis for the demonstration, Section 5 introduces necessary background work. Section 6 demonstrates our methodology, and as follows, Section 7 presents performance metrics and retrieved images from random queries. Section 8 introduces the new benchmark "TIR", how to collect and compile it, and characteristics of it. Retrieved images from random queries is also shown in this section.

2
LITERATURE REVIEW

Multimodal learning is a wide research area in Deep Learning for last decades. There are so many studies are proposed and explored which have different objectives, sub-tasks and more. As a result of high-speed developments in both Natural Language Processing and Computer Vision, vision-and-language interaction has gained more attention. Taking into account the uncountable number of developments in this area, we would like to divide past and current reserch as **generative** models and **non-generative** models.

Generative models mainly include a sequence-to-sequence architecture, to generate image from text or vice-versa. One of the beginnings of sequence-to-sequence architectures, [13] proposed a multimodal recurrent architecture. They use a Recurrent Neural Network (RNN) to extract language features, and a Convolutional Neural Network (CNN) to generate images from text embedding. They adapt this model for both generation and ranking for retrieval. The same approach is used in [14], however they replaced Recurrent Neural Network with Long Short-Term Memory Network (LSTM), which gives embedding interaction spaces with a better precision. Following the developements on Machine Translation with alignment of source and target languages with cross-attention ([15], [16], [17]), and visual attention ([18], [19]), the first cross-attention model between two modalities is proposed by [20]. They use a CNN layer to extract image embeddings and RNN layer to generate captions. The main contribution is the adaptation of hard and soft attentions, which are proposed in [17], to learn visual-language embeddings and alignments. One of the first joint embedding space model is proposed by [21]. They use a R-CNN like model to generate captions not just for whole image but also for arbitrary image regions. With the latest developments on text transformers [11] and image transformers [22], multimodal generation can be accomplished by a full transformer network, as shown in [23].

On the other side, there are several non-generative models are proposed, with objectives like image-text matching or learning joint embedding space. As a multimodal distributional model, authors of [24] use align basic vector representation of text classes and SIFT for image features. With the same motivation, [25] use SIFT descriptors and LDA topic modeling to learn alingments with Jensen Shannon divergence. To explore zero/few shot capabilities of multimodal embeddings, [26] approaches to problem as metric learning task which is accomplished by minimizing a distance metric between image embeddings and text embeddings. We encourage or readers to read [26] in detail to understand our deep metric learning based retrieval problem in Section 6. Regarding a novel architecture, authors of the VL-BERT [27] show that a Transformer network for aligning image and text features outperforms previous works. They extracts RoI features with Faster-RCNN [8] model, then passes this features with text features to a Transformer Encoder. To differentiate modalities, they introduced a segment embedding layer, where each modality has unique indentifier embedding. The pre-training tasks for VL-BERT are MLM and Masked RoI Modeling, where a random RoI is masked and predicted with the natural language supervision. With more robust pre-training tasks, authors of LXMert [28] uses a Transformer Encoder, where modality inputs are RoI features, Position Features and word embeddings, index embeddings. Each modality has a unique second pass feature extractor layer wich is a single Transformer Encoder. Then, outputs of each modality encoder concatenated and passed to cross-modality Transformer Encoder. For LXMert, four pre-training task is proposed: MLM, Masked Object Prediction, Image-Text Matching and Visual Question Answering. We encourage or readers to read [28] in detail to understand separate modality encoders with cross-modal image-text matching encoder in Section 6. A full Transformer network for aligning both modalities is introduced in [29], namely ViLT. With a basic formulation, they pre-process image as in ViT model [22] (introduced in Section 5) and passes it with text token ids to single Transformer Encoder. They use three pre-training objective: Image-Text matching as in LXMert, MLM and Word Patch Alignment.

A full Transformer model with metric learning objective is proposed in [12], namely CLIP. They use two distinct Transformer Encoder layer for text and image (ViT). To project the emebddings to the same space, they use a projection head which is a combination of Layer Normalization and Feed Forward networks. To align modalities, they maximize the dot product of image embeddings and text embeddings, by calculating loss with symmetric cross entropy loss. The advantage of this formulation is that retrieval with embedding similarity ranking becomes unbiased. Also, using only cosine similarity is faster than a Transformer Encoder, when it comes to optimizing the latency. Besides, it is portable to use it with efficient search algorithms such as faiss [30].

3
FEASIBILITY

In this chapter, we are going to define collection of feasibilities, which are one of the important building blocks of this thesis. Feasibilities are classified as Technical Feasibility (software and hardware), Schedule Feasibility, and Legal Feasibility.

3.1 Technical Feasibility

In this section, our main concern is to define both software and hardware feasibility, for reproducibility and clearness.

3.1.1 Software Feasibility

Most of the Deep Learning research is done with Python programming language; due to its effectiveness, ease to use and accelerated compatibility with C++ and CUDA (i.e., binding). Moreover, when the training dataset and number of parameters of the model increases; this computation should be done in fastest but most practical way. Consequently, in recent years number of open source libraries/frameworks; such as PyTorch, DyNet, JAX, are increased. We thus choose Python programming language to implement our models.

In spite of the effectiveness of Python, such libraries and hardware must be perfectly connected. In other words, each library comes with a dependency, which should be specified in clearest way. Each version of a Python package is able to run with different hardware and/or different version of an another Python package. Here, we explain most of the libraries and hardware properties which are used, in Table 3.1.

3.1.2 Hardware Feasibility

A deep learning model has million or billion number of parameters to derive and tune. Hence, to train a deep learning model on GPU with CUDA compatibility is more computationally efficient than a CPU. We use Nvidia P100 with CUDA version of 11.4,

Table 3.1 Python Packages and Version Dependencies

Package	Version
PyTorch	1.9.1
Hugging Face Transformers	4.18.0
pandas	1.3.5
numpy	1.20.3
albumentations	1.1.0
timm	0.5.4
PIL	8.2.0

two NVIDIA 2080TI with CUDA version of 10.1 and NVIDIA 1060 with CUDA version of 9.1.

3.2 Legal Feasibility

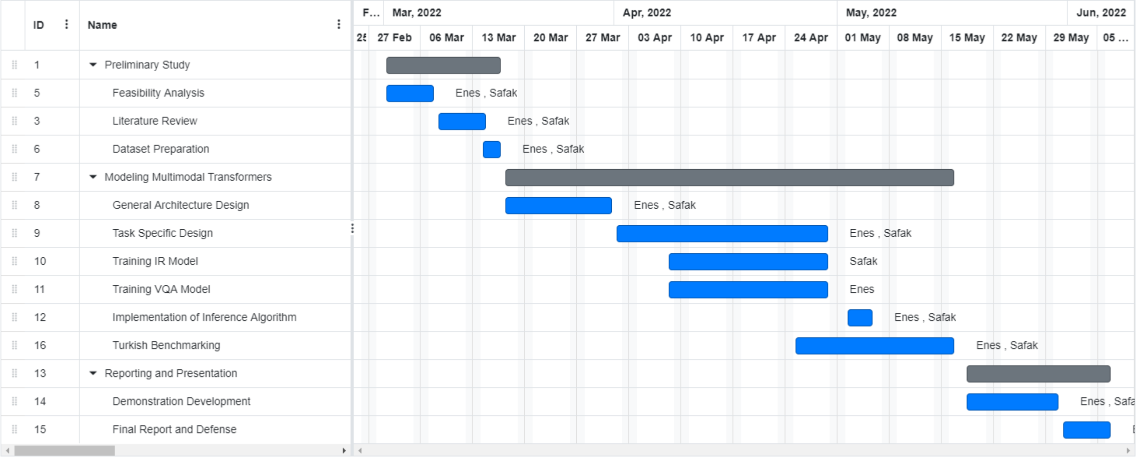
Since the software we use is open source, it does not create any legal liability. The benchmarks that we use for training and inference are shared as free with registration.

Table 3.2 Licenses of Packages

Package	Version
PyTorch	BSD
Hugging Face Transformers	Apache License 2.0
pandas	BSD-3-Clause License
numpy	BSD-3-Clause License
albumentations	MIT License
timm	MIT License
PIL	HPND License

3.3 Economic Feasibility

Since the software we use is open source, there is no economic feasibility conditions. We use free but limited GPUs in Kaggle, Google Colaboratory. Other GPU hardwares belong to this thesis' students which do not require any economic conditions.



ID	Name
1	Preliminary Study
5	Feasibility Analysis
3	Literature Review
6	Dataset Preparation
7	Modeling Multimodal Transformers
8	General Architecture Design
9	Task Specific Design
10	Training IR Model
11	Training VQA Model
12	Implementation of Inference Algorithm
16	Turkish Benchmarking
13	Reporting and Presentation
14	Demonstration Development
15	Final Report and Defense

Figure 3.1 Gantt Diagram

3.4 Workforce and Time Feasibility

This thesis is written by two senior student and planned to finish in one academic term. It consist of four main parts: literature and dataset review, modeling multimodal Transformers, modeling inference and prediction, and releasing the first benchmark for Turkish Image Retrieval and Visual Question Answering.

4

SYSTEM ANALYSIS

The demonstration of the topic is divided to two main sections: retrieval (for both image and text), and visual question answering. To focus on retrieval, first, the user selects a database, which contains images or texts. As an example, then, the user passes a query to search engine to get relevant images which are sorted by ranking model. The number of retrieved images is varied and chosen by the first user. The design and contents of the database can be pre-defined ar post-defined by the user.

Another system usage contains visual question answering. The user selects or uploads an arbitrary image to the system, then asks the relevant question/s to the system. With conditioning the image, retrieval system gives most relevant answer. The design of the number of relevant answers varies and can be chosen by the first user. If multiple answers are desired, then the sistem gives answers with ascending order regarding to the confidence levels.

It is planned to implement the inference system with GPU acceleration. If user chooses CPU for inference, latency is hurt as expected.

4.1 Use Case Diagram

The use case diagram of above scenario is shown in Figure 4.1.

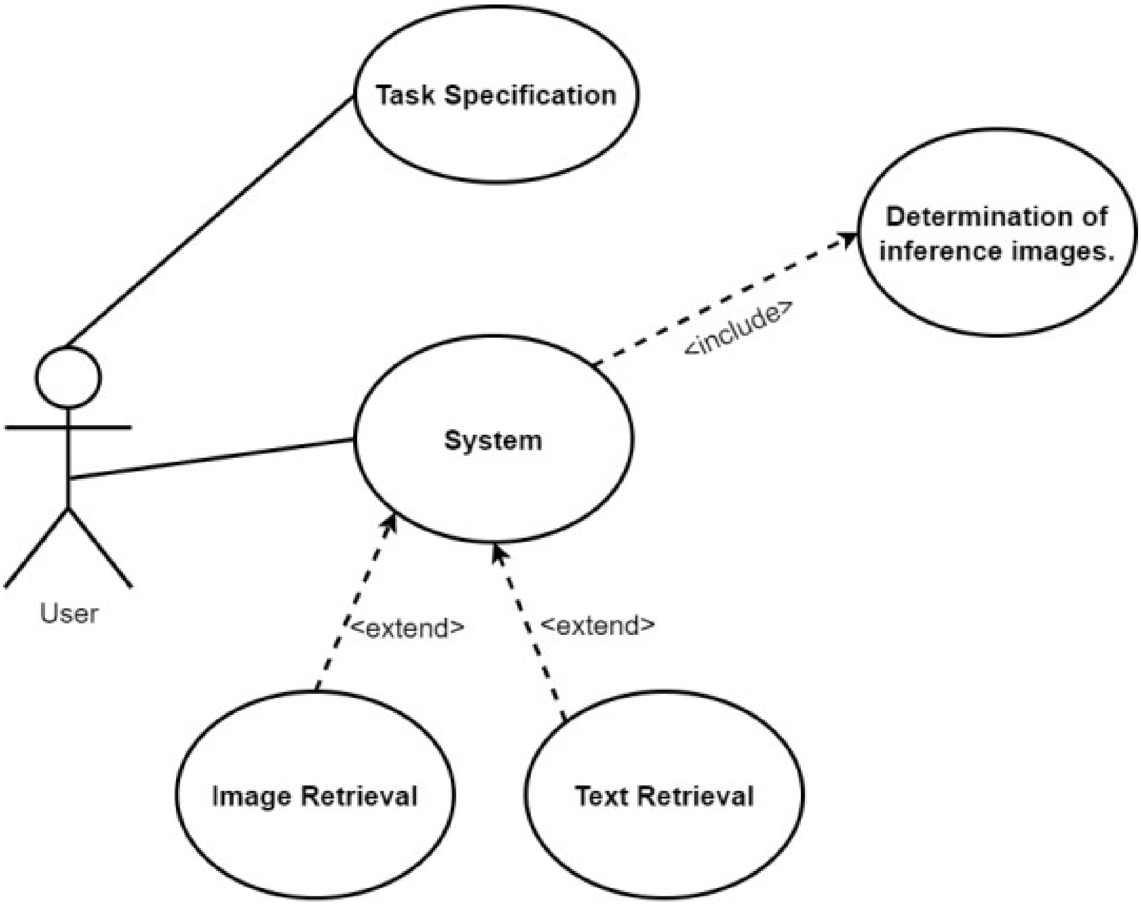


Figure 4.1 The Use Case Diagram

5
BACKGROUND

Following the latest developments on Natural Language Processing and Computer Vision models, we aim to use Transformer [11] based models to actualize our study. Since 2017, Transformers are accepted as state-of-the art model in various Natural Language Processing tasks, such as; Machine Translation, Summarization, Sequence Classification, Sequence Labeling, Question Answering etc. Besides all these developments, GPU manufacturing and amount of labeled data increasing exponentially. Over last three years, Transformer based models are adapted to Computer Vision tasks such as Object Detectoin, Image Classification, Image Segmentation etc., and achieved state-of-the-art performance on several benchmarks.

These developments caused the Transfer Learning, self-supervision, and pre-training of Transformer based models. As a result, fine-tuning a pre-trained model on a downstream task becomes more data and computationally efficient, even in few or zero shot degree.

Since our study comprises multimodal learning, we use pre-trained language and vision transformers, and fine-tune them with a multimodal interactor objective for various tasks such as Retrieval, Ranking, Visual Question Answering, Zero Shot Image Classification.

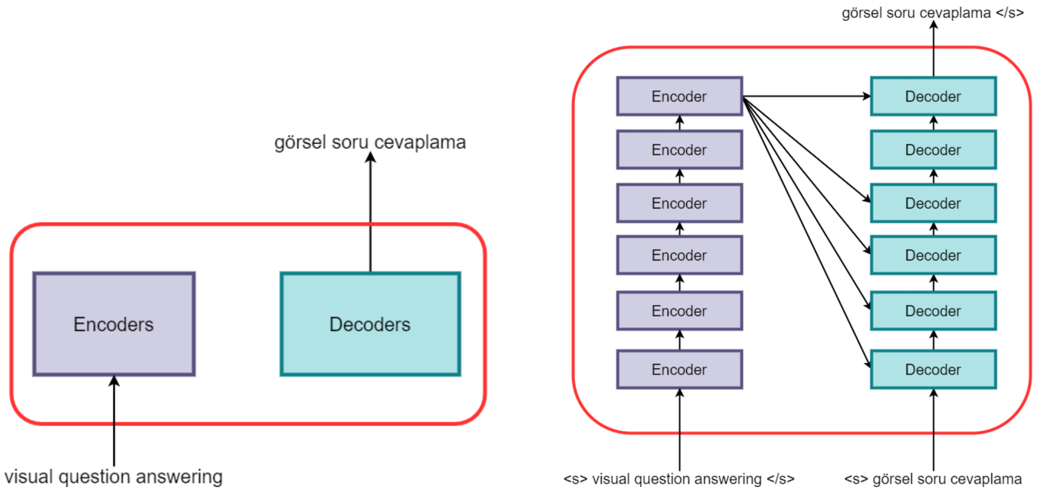
This section provides theoretical aspects on elements of the study and explains in detail.

5.1 Transformers

Transformer [11] is a architecture for contextualized word emebddings. Contextualized word embeddings are mutable. In skip-gram based word embedding models, such as word2vec or GloVe; word embeddings are immutable. This means, the vector of a word never changes. However, contextual embeddings provides mutable word vectors, they are changing by the context of the sentence. As Recurrent

Neural Networks (RNNs), Transformer is a sequence-to-sequence model as well. However, it is purely built on attention mechanism. Besides, all calculations happen at once. Hence, it is more parallelizable and requiring significantly less time to train.

Transformer has encoder and decoder. The encoder part of the Transformer is bidirectional, the decoder part of the Transformer is unidirectional. To compare with seq2seq RNN, we pass the source sentence to Transformer's encoder, and Transformer's decoder decodes encoder's output to target sentence. The encoding component is a stack of encoders. The decoding component is a stack of decoders of the same number.



(a) Folded Transformer **(b)** Unfolded Transformer

Figure 5.1 The Transformer.

The inside components of encoder and decoder are nearly the same. The encoder learns the important "intra-features" of input sentence and the decoder also learns the important "intra-features" of output sentence, however, it learns cross-alignments of input and output as well. This learning procedure is done by self-attention.

5.1.1 Self-Attention

Self-attention computes similarities inside a sentence. Consider the sentence: "The dog was barking. Seems like it is hungry.". There is no doubt that word "dog" and "it" represent same object. Self-attention extracts these relationships. The formulation of the self-attention is based on a learned version of cosine similarity.

Consider that you have a collection vectors of n words $\mathbf{V} \in \mathbb{R}^{n \times D}$. Select an arbitrary word i, and calculate dot product between each other word vectors:

$$s_{ij} = v_i \cdot v_j \tag{5.1}$$

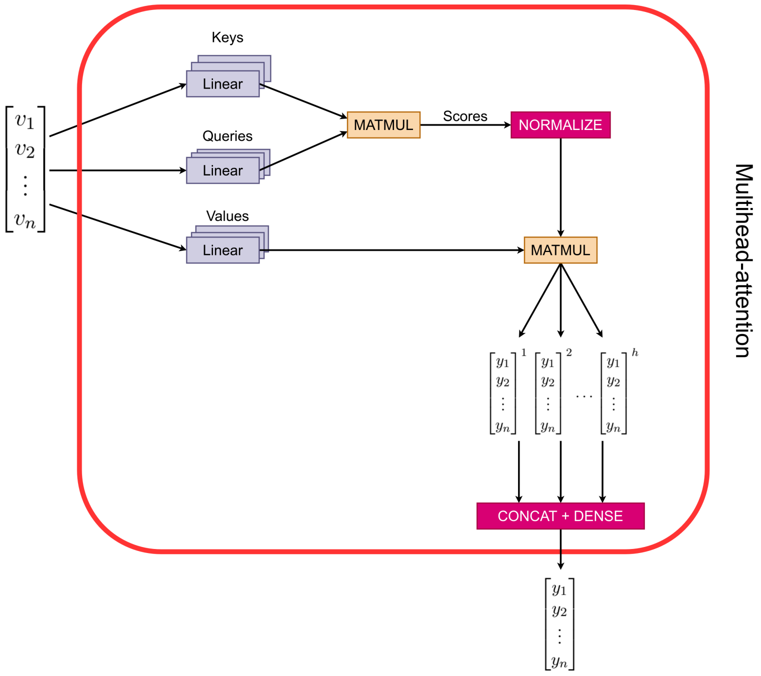

Keys
Linear
Queries
Linear
Values
Linear
v_1 v_2 $\vdots$ v_n
MATMUL
Scores
NORMALIZE
MATMUL
y_1 y_2 $\vdots$ y_n 1
y_1 y_2 $\vdots$ y_n 2
…
y_1 y_2 $\vdots$ y_n h
CONCAT + DENSE
y_1 y_2 $\vdots$ y_n
Multihead-attention

Figure 5.2 Self-attention block

Each scalar s_{ij} represents similarity between words v_i and v_j. However, each similarity scalar s_{ij} is independent from other values. We thus normalize them

$$w_{ij} = \frac{s_{ij}}{\sum_{k=0}^{n} \sum_{l=0}^{n} s_{kl}} \tag{5.2}$$

After normalization, each w_{ij} tells us the relationship between word vector v_i and v_j. After this process, v_i can be re-written in terms of all v_js, which corresponds to contextual representation of word vector v_i:

$$y_i = \sum_{k}^{n} v_k \cdot w_{ik} \tag{5.3}$$

The intuition of it that each vector is multiplied by its relevance on v_i, which is represented by s_{ij}, then summed. Above formulations are done on a single word vector, however, it can be done in a matrix level

$$\mathbf{S} = \mathbf{V} \cdot \mathbf{V}^T \tag{5.4}$$

$$\mathbf{W} = softmax(\mathbf{S}) \tag{5.5}$$

$$\mathbf{Y} = \mathbf{W} \cdot \mathbf{V} \tag{5.6}$$

If we inject this "re-writing" system into Transformer, we must add learnable parameters. Instead of using single **V** matrix; Query (**Q**), Value (**V**) and Key (**Q**) matrices, which are the linear projection of input embedding, are introduced under the equations Eq (2.4) and Eq (2.6):

$$\mathbf{A} = softmax\left(\frac{\mathbf{Q}\cdot\mathbf{K}^T}{\sqrt{d_Q}}\right)\cdot\mathbf{V} \tag{5.7}$$

If each Q, K, V is introduced *h* time in an ensemble manner, it is called that "multihead self-attention".

5.2 Pre-Trained Language Models

After ELMo [31] and Transformer model is introduced, first pre-trained Transformer model is came out in [9], named BERT. BERT achieves state-of-the-art performances over ten downstream tasks and is widely used on both research and industry. After BERT model, there are several pre-trained language models are proposed. Some of them are seq2seq language models [32], [33]; encoder based language models [9]; [34], and decoder based language models [35].

In this section, seq2seq and autoregressive pre-trained language models are not discussed.

5.2.1 BERT

BERT [9] is a language model, which is a stack of only Transformer's encoder layers. Trainin procedure of BERT is two staged:

1. Pre-training on a large corpus with a pre-training objective.
2. Fine-tuning on a downstream tasks with task specific modifications.

Recent work on word embeddings [36], [37] before BERT produces immutable word vectors for each word, ignoring the its contextual meaning in sentence. Benefits of self-attention based language model include contextual word vector producing, which is mutable in context. These mutable word vectors are called as "contextual representation".

At pre-training stage, BERT's objective is to predict masked tokens in input, called Masked Language Modeling (MLM); and Next Sentence Prediction (NSP). This objective is done in self-supervised fashion. BERT thus does not need a labeled corpus for pre-training. This corpus is splitted into non-overlapping segments with length of 512 tokens (typically $N = 512$), and 15% of tokens in each segment is masked

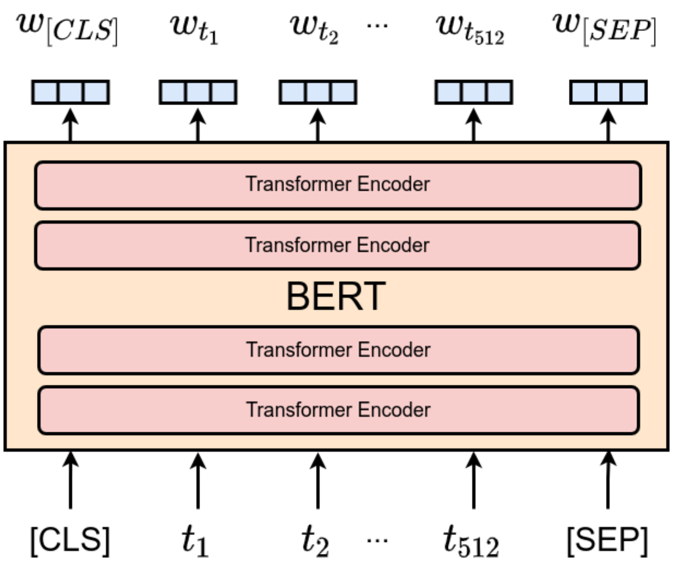


Figure 5.3 BERT Model

with `[MASK]` token randomly. Besides the `[MASK]` token, several special tokens are used in BERT: `[CLS]` is added to start of the segment and `[SEP]` is added to end of the segment (also used for splitting different segments in such tasks like textual entailment).

The objective is to maximize log-likelihood of observing masked tokens $\tilde{\mathbf{x}}$ with given masked input segment $\hat{\mathbf{x}}$ for sequence length N

$$\max_{\theta} \log p_{\theta}(\tilde{x} \mid \mathbf{x}) \approx \sum_{t=1}^{N} \mathbb{1}(x_t) \log p_{\theta}(x_t \mid \hat{\mathbf{x}}) \tag{5.8}$$

where θ is model's parameters, $\mathbb{1}(x_t)$ is a indicator function that gives 1 for $x_t = \texttt{[MASK]}$, otherwise zero [38].

Fine-tuning of BERT is different from pre-training procedure. It is done with a task specific modification of pre-trained BERT model, typically a task specific layer is added to top of the BERT. It can be fine-tuned for Question Answering, Sequence Classification, Sequence Labeling etc., however we are going to introduce only sequence classification tasks, due to it is our significant building block for both retrieval and VQA.

In fine-tuning, main parameters of BERT model are freezed typically. In other word, pre-trained parameters are not updated during fine-tuning stage. If the task of sequence classification is considered, a fully connected layer top of the contextual vector of `[CLS]` is generally used. The motivation of this is `[CLS]` is "crossed" by all other tokens due to bidirectionality of BERT, and does not have positional bias.

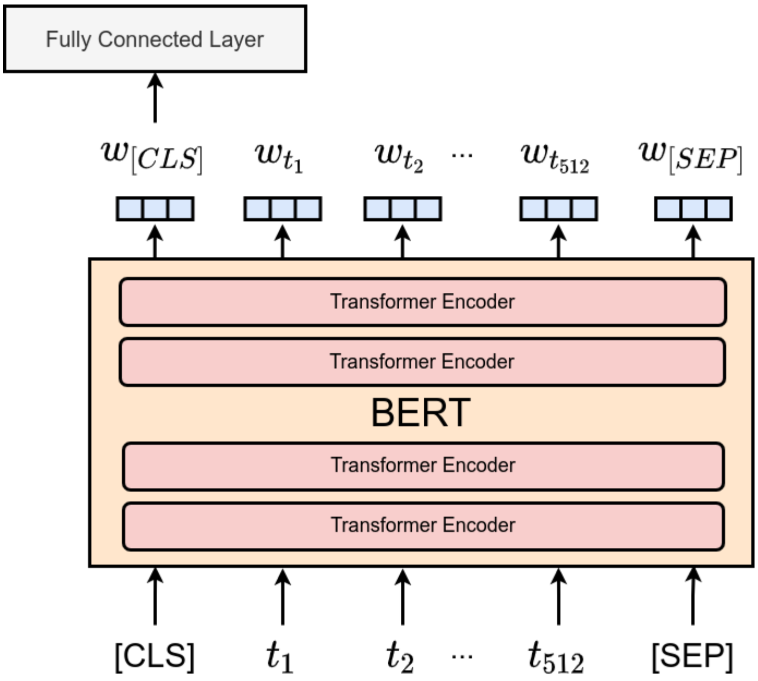


Figure 5.4 Fine-Tuning Modification of BERT for Sequence Classification

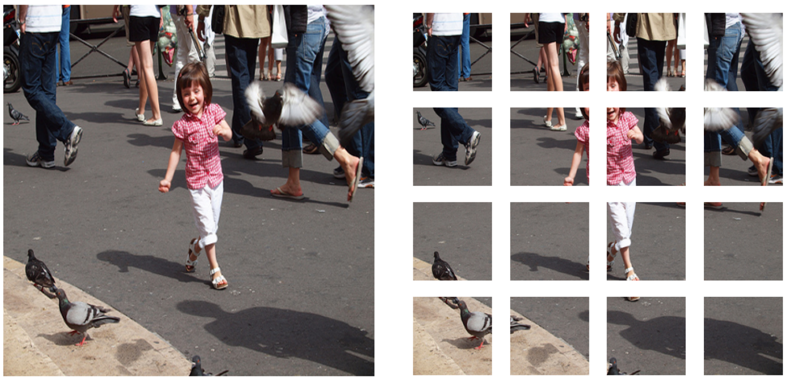
Besides, it is the most computational efficient way, comparing to concatenating all word vectors or pooling over all word vectors.

5.3 Vision Transformer

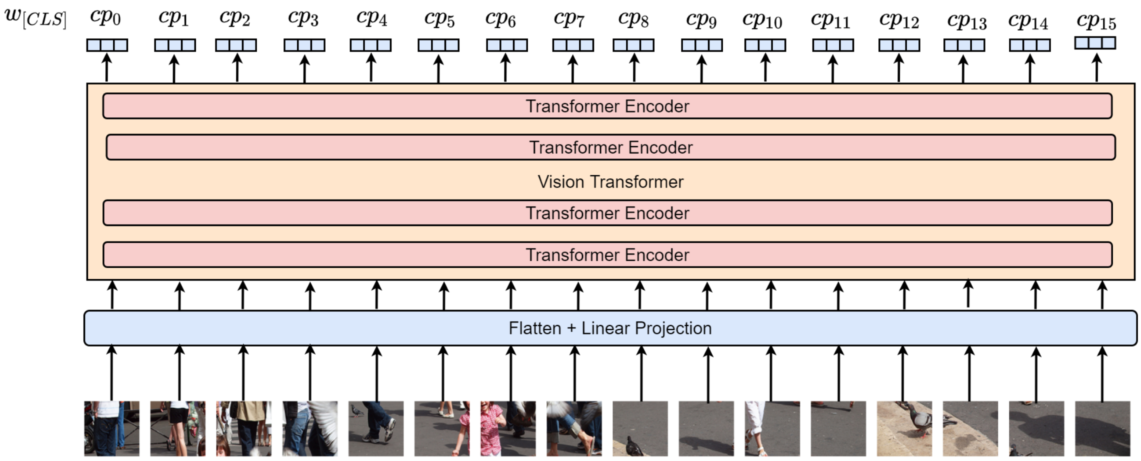
Developments over Transformer based pre-trained language models leads up Vision Transformers. It is used for Object Detection [39], Image Segmentation [40]. Consequently, pre-training of Vision Transformers is proposed in [22], namely ViT. ViT model is pre-trained on large benchmarks, such as ImageNet, with a supervised objective like classification or self-supervised objective Masked Patch Prediction.

First of all, an image $I \in \mathbf{R}^{W \times H \times C}$ is resized to a square image $I' \in \mathbf{R}^{M \times M \times C}$. Then, this square image I' is splitted into N patches, with dimension of $P \in \mathbf{R}^{L \times L \times C}$, where $L = M/N$. As a prior, this motivation tells us that we are interested with relationship among patches rather than single pixels. These paches are flattened with a single resizing operation and each patch passed to a linear projection layer, such as a single feed forward layer or a CNN. Thus, each pixel has a position $pos \in \{0, ..., N\}$.

As in BERT, model structure of Vision Transformer is a stack of Transformer Encoder layers. Each encoder computes the self-attention between Query and Key vectors, which are simply linear projection of flattened patches. At the output, contextual representation of each patch is obtained. Vision Transformer also introduces a special



(a) Input Image **(b)** Patches



(c) Vision Transformer

Figure 5.5 Vision Transformer Model

patch for `[CLS]`, which can be used for classification. To predict the class labels for supervised pre-training objective, a single fully connected layer is integrated to top of the contextual vector of `[CLS]`.

6

SYSTEM DESIGN: DEEP MULTIMODAL LEARNING WITH VISION-AND-LANGUAGE TRANSFORMERS

In this section, several improvements over Multimodal Learning with Transformers is introduced. We experiment two distinct tasks, Retrieval and Visual Question Answering. Each task and model is explained in detail and evaluated on different benchmarks.

A multimodal model should learn visual representations with a generalized natural language supervision; in other words, an image and a text should be aligned perfectly. This task can be done in various ways. For example, visual and language representations can be learned in a joint space, by minimizing a distance metric $d(I,T)$ where I is a image embedding and L is sentence embedding. Similarly, instead of a similarity distance metric $d(\cdot,\cdot)$, minimization of clustering loss brings closer each relevant embeddings from different modalities. Another way to align is to matching images and texts as self-supervised classification objective. These objectives are explained in relevant sections, in detail.

Another problem in multimodal learning is to extract distinct features from a modality, for example RoI of an image or dependencies inside a text. This suggests that, we have to extract *intra* and *inter* features from modalities. On account of this, we modify and use Transformer model as a multimodal structure. In general, *intra* feature extractor Transformer is choosen separately: a unimodal text Transformer for language, a unimodal image Transformer for vision. As an *inter* fetaure extractor, typically, a "multimodal interactor" layer is used. Since Transformer is a stack of self-attention layers, we believe that is will be the most proper way to extract these features. Choice of this multimodal interactor varies from task to task. This layer generally aligns and find similarities/connections between different modalities, image and text.

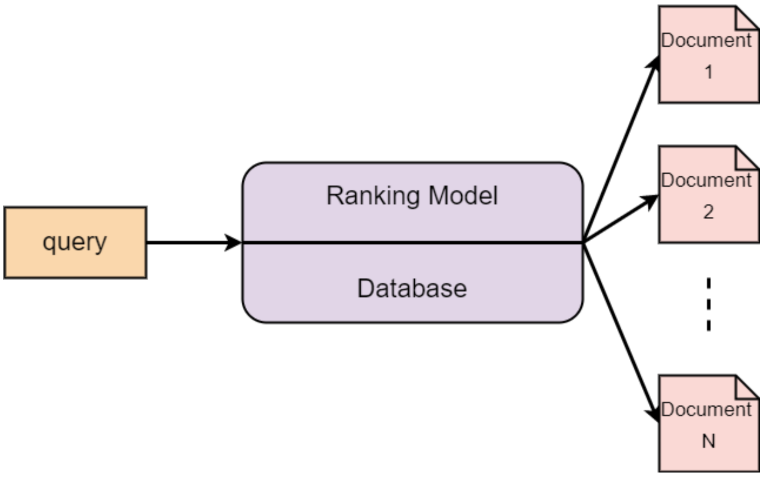


Figure 6.1 Document Retrieval

6.1 Retrieval and Ranking

Before going deeper into the details, it should be described in clearest way, "what is retrieval?" and "what is the motivation behind retrieval?". Information Retrieval (IR) is a task for collecting relevant documents from a query or collection of queries. When a query is passed to search engine, all documents are retrieved with a ranking score, which tells us the relevance between the query and the document. This procedure can be done in various ways. For example, PageRank [41] produces a likelihood function that parameterized the randomness of click of a user. Or, with a proper indexing, simple TF-IDF model may work for simpler queries and homogeneous documents. When it comes to complex queries and heterogeneous documents, probabilistic models (as explained above) do not work well due to high bias.

After developments on word vectors and knowledge graphs [42], area of information retrieval had gained the power and benefited from these. Training of word embedding models on large documents and queries strengthen the ranking metrics. As in intrinsic evaluations, similarity of query vectors and document vectors can be calculated with a simple clustering or distance metric (for example Euclidean Distance). Likewise, a graph can be initialized with documents, learned with similarities, and search can be done with traversing the graph.

Another problem in retrieval is to rank the images from a query or a collection of queries. Nonetheless, for non-complex images and queries, a simple histogram matching model works fine. However, with regard to complex images and queries, the task of retrieval based on pixels becomes ambiguous and ineffective. Hence, the modeling of image (also text retrieval from an image) retrieval should be done and described with highest precision.

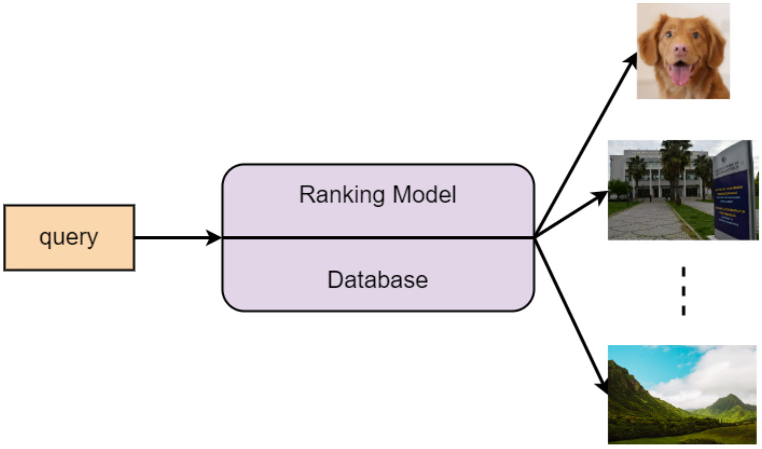


Figure 6.2 Image Retrieval

We follow two different objectives: image-text matching and deep metric learning on different benchmarks. We report our results on recall-at-k (R@k) evaluation metric and show example images/texts with relevant query/image.

6.1.1 Image-Text Matching

The first objective choice is named as "image-text matching". We experiment this objective on Flicker30k dataset, which is going to be described in detail, later. This objective is built on only Transformer based model: two distinct *intra* feature extractor (for both image and text) and a *inter* feature extractor, which is called "multimodal interactor". In this section; details of these objective, model, results and image retrieval examples are shown.

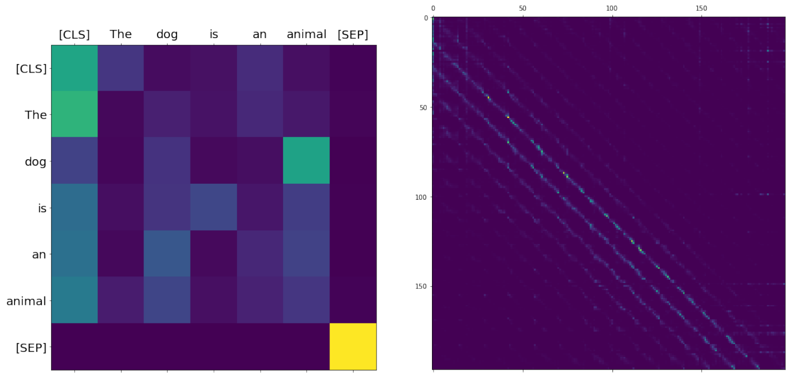
For *intra* feature extractor for text modality, we use DistilBERT model, which is a knowledge distilled version of original BERT model. DistilBERT is 40% smaller than BERT. Tokenizer of DistilBERT is word-piece tokenizer which is a subword tokenizer to eliminate OOV (out-of-vocabulary) words, narrow the vocabulary size, handle morphological rich languages (i.e., Turkish). Consider following sentence

"Multimodal Transformers are capable to align."

The word-piece tokenizer produces tokens as follows

["Multi", "mod", "al", " ", "Transform", "ers", "are", "capable", "to", " ", "align", "."]

DistilBERT produces an output tensor with dimensions of $B \times L \times 768$ where B is the batch size, L is the maximum sequence length in the batch, and 768 is the dimension



(a) DistilBERT **(b)** ViT

Figure 6.3 Attention Matrices in DistilBERT and ViT. Attention layer and heads are choosen randomly.

of each word vector. For tasks like single sentence classification, DistilBERT has special `[CLS]` token as well.

For *intra* feature extractor for text modality, we use Vision Transformer (ViT). Passing images to ViT model, each image is pre-processed to square image with dimension of $3 \times 224 \times 224$. Then, each square image is separated to patches with $C \times 16 \times 16$. This means, we have 197 (+1 for special token `[CLS]`) total number of patches. Each patch is flattened and then passed to the Vision Transformer.

As shown in Figure 6.3 (a), DistilBERT's self-attention layer is capable to dependencies in input text, for example, token "dog" attends to token "animal" in row attention. Figure 6.3 (b) also shows that ViT has mostly sparse attention matrix, it generally attends to diagonal and bidirectional dilated positions.

Output of DistilBERT and ViT represents each modality with contextual information, however, interactions among modalities have not been extracted yet. To address this problem, we designed a multimodal interactor layer. This layer is a stack of self-attention layers with non-casual masking (Transformer's Encoder).

First of all, the output of ViT and DistilBERT are concatenated to a single tensor, which have dimension of $B \times (L + 197) \times 768$. However, the position of `[CLS]` vector of ViT relocated to right of the `[CLS]` vector of DistilBERT:

$$X' = [v_{DistilBERT_{CLS}}; v_{ViT_{CLS}}; v_{DistilBERT_0}; \ldots; v_{DistilBERT_L}; v_{DistilBERT_{SEP}}; v_{ViT_0}; \ldots; v_{ViT_{197}}] \tag{6.1}$$

this formulation give same performance with averaging CLS vectors, however we do not want to reduce the alignment of two modalities. Then, the linear projection of

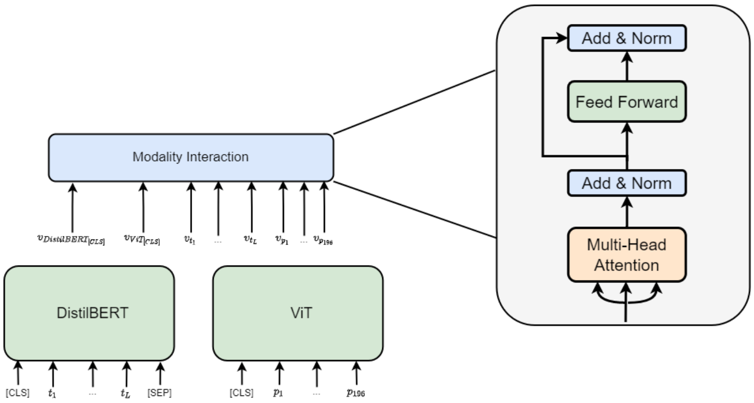


Figure 6.4 Multimodal Transformer Architecture

each vector is obtained, namely Query (Q), Key (K), Value (V) vectors

$$\mathbf{X}' \cdot \mathbf{W}_Q + POS_{X'} = \mathbf{Q} \tag{6.2}$$

$$\mathbf{X}' \cdot \mathbf{W}_K + POS_{X'} = \mathbf{K} \tag{6.3}$$

$$\mathbf{X}' \cdot \mathbf{W}_V + POS_{X'} = \mathbf{V} \tag{6.4}$$

where POS is the positional encoding for input vectors. Then, alignment of each modality is calculated with self-attention

$$\mathbf{A} = \left(\frac{\mathbf{Q} \cdot \mathbf{K}^T}{\sqrt{d_{\mathbf{Q}}}} \right) \cdot V \tag{6.5}$$

Besides that, the non-casual mask is relocated to concatenated input vector. Normally, non-casual mask for DistilBERT is masking the `[PAD]` positions, however, when the both modality vectors are concatenated, non-casual mask must be relocated for text vector.

At training, with probability $p = 0.5$, image of a single caption is changed. The objective is to classify whether it is changed or not. This states that, retrieval task can be done with a binary classification. Thus, it is needed that a single classifier layer top of the output of $[v_{DistilBERT_{CLS}}; v_{ViT_{CLS}}]$ embedding. This embedding has information of both multimodality and alignment of each modality, as explained in Chapter 5.

$$\mathbf{Z} = LN(\mathbf{A}) \tag{6.6}$$

$$p = softmax(\mathbf{Z}_0^N \cdot \mathbf{W}_{pool}) \tag{6.7}$$

where LN is Layer Normalization.

The multimodal interactor encoder layer has three stack of self-attention layers, with four heads. The dimensions of Query, Key and Value vectors are set to 256 and the inner embedding dimension is set to 128. The dimension of contextual vectors at output layer is set to 768. We use Adam optimizer [43] with $\beta_1 = 0.9$, $\beta_2 = 0.999$. At training time, we apply label smoothing [44] with $\epsilon_{ls} = 0.1$.

Batch size of training data is set to 32 and test data is set to 64. The initial learning rate is 2e-5 and we use cosine annealing with linear warmup.

Image-text matching model is trained with Flickr30k dataset [45]. Flickr30k contains 31000 images, each image has nearly five captions, which are collected from Flicker. We use same train-test split in [21]. Each image from Flicker has no special domain, it represents general real life images; such as people, animals, events, and more.

6.1.2 Deep Metric Learning for Retrieval

The latter objective choice is named as "Deep Metric Learning for Retrieval", which is nearly same as CLIP model [12]. We experiment this objective on Flicker30k dataset, which is going to be described in detail, later. This objective is built a Transformer based model for extracting *intra* features of texts and a ResNet model for extracting *intra* features of images. Rather than a multimodal interaction layer, the objective is to represent these output vectors in a joint space and maximize their similarity $d(image, text)$. In this section; details of these objective, model, results and image retrieval examples are shown.

To extract the text features DistilBERT model is used as well as in image-text matching model. To extract the image features, we use pre-trained ResNet50 model [46]. The motivation of usage if ResNet50 is reducing the latency per query when the retrieval is done at inference phase, since ViT has more parameters than ResNet50 model. ResNet50 model produces output vector with dimension of $B \times 2048$.

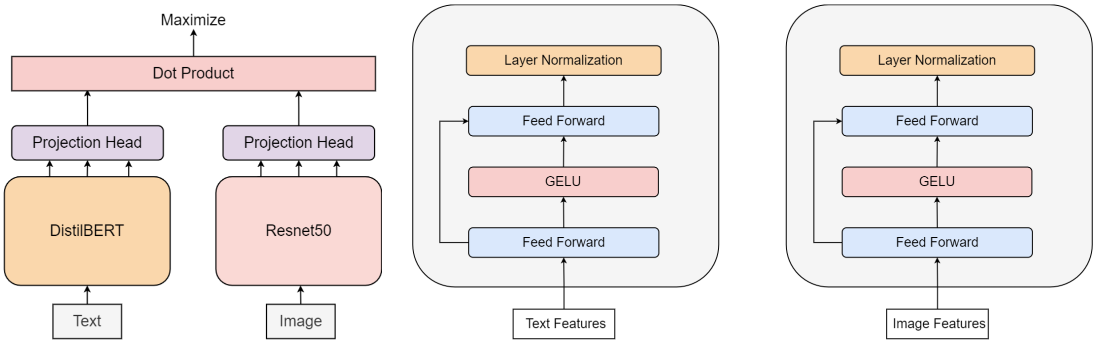
To model a deep metric learning architecture, a projection head is designed to project text and image features vectors to same dimensionality. This projection head includes feed forward layers and a layer normalization layer with residual connection between input embedding.

$$\mathbf{Z}_1 = \mathrm{MLP}(\mathbf{V}_{\mathrm{modality}}) \tag{6.8}$$

$$\mathbf{Z}_2 = \mathrm{GELU}(\mathbf{Z}_1) \tag{6.9}$$

$$\mathbf{Z}_3 = \mathrm{MLP}(\mathbf{Z}_2) + \mathbf{Z}_1 \tag{6.10}$$

$$\mathbf{E} = \mathrm{LN}(\mathbf{Z}_3) \tag{6.11}$$



(a) Model **(b)** Projection Heads

Figure 6.5 Modeling Deep Metric Learning for Retrieval

Now, we call output of image projection head as "image embeddings" and output of image projection head as "text embeddings". The architecture of projection head is shown in Figure 6.7.

After output of the projection layer for both image and text, the dot product of both embeddings is calculated as

$$\mathbf{s} = \mathbf{E}_{\text{text}} \cdot \mathbf{E}_{\text{image}}^{T} \tag{6.12}$$

The dimension of output in dot product layer is $B \times 1$, which contains similarity scalars between image embeddings and text embeddings. Deep Metric Learning plays a role when our aim is to maximize this similarity vector $\mathbf{s}$. When the loss is calculated for dot product, gradient signal is flowing through to both text and image projection head by backpropagating. Hence, the embeddings for both modality is learned in a joint space. This allows us to compute similarities between query and images when we are trying to retrieve relevant images. The training algorithm of the model can be formulated as To train the model, batch size is set to 32 for both training and testing.

Algorithm 1 Training Deep Metric Learning Model for Retrieval

```
Require: batch $B$, temperature $t$
$I \leftarrow \text{ResNet}(B[\text{"image"}])$
$T \leftarrow \text{DistilBERT}(B[\text{"text"}])$
$E_I \leftarrow \text{ImageProjectionHead}(I)$
$E_T \leftarrow \text{TextProjectionHead}(T)$
$l \leftarrow E_T \cdot E_I^T$
$\text{sim}_I \leftarrow E_I \cdot E_I^T$
$\text{sim}_T \leftarrow E_T \cdot E_T^T$
$\text{targets} \leftarrow softmax\left(\frac{\text{sim}_I + \text{sim}_T}{2 \times t}\right)$
$\text{Loss}_{\text{text}} \leftarrow \text{cross\_entropy}(l, targets)$
$\text{Loss}_{\text{image}} \leftarrow \text{cross\_entropy}(l^T, targets^T)$
$\text{Loss} \leftarrow \frac{\text{Loss}_{\text{text}} + \text{Loss}_{\text{image}}}{2}$
return Loss
```

Weights of pre-trained DistilBERT and ResNet50 is updated during fine-tuning. The learning rate for DistilBERT is set to 1e-4 and ResNet50 is set to 1e-5. Learning rate for both projection heads is set to 1e-3. The model is trained for 4 epcohs with AdamW optimizer [47], betas are set to $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay is set to 1e-3. As learning rate scheduler, ReduceLROnPlateau is used with patience of 1 and factor of 0.8.

Comparing the two objectives, it is inevitable that metric learning objective outperforms the image-text matching objective. Because, each feature extracting layer is updated during fine-tuning phase of the model, and representing images and texts in a joint space gives us the unbiased estimation strength of the cosine similarity. In inference time, retrieval can be formulated with following algorithm

Algorithm 2 Inference with Cosine Similarity

```
Require: image database $D$, query $Q$, tokenizer, $k$
token_ids ← tokenizer.encode($Q$)
$T$ ← DistilBERT(token_ids)
$E_T$ ← TextProjectionHead($T$)
$I_n$ ← []
for batch $B$ in $D$ do
    $I$ ← ResNet($B$["image"])
    $E_I$ ← ImageProjectionHead($I$)
    $I_n$.append($E_I$)
end for
$I_n \leftarrow \frac{I_n}{||I_n||_2}$
$E_T \leftarrow \frac{E_T}{||E_T||_2}$
$S \leftarrow E_T \cdot I_n^T$
retrieved_images ← sort($S$)[:$k$]
return retrieved_images
```

First of all, query is encoded to token ids and passed to DistilBERT and text projection head to get image embeddings. After that, embedding of each image is calculated with ViT and image projection head, then stored as an array. The next step is to normalizing the distribution of each image and query embedding with L2 normalization. Last of all, cosine similarity is calculated between image embeddings and query embedding.

6.2 Visual Question Answering

The definition of Visual Question Answering (VQA) is an important case in Natural Language Supervision. As mentioned in retrieval, it is a major problem that image and text modalities must be learned and aligned flawlessly. If we examine this problem for VQA, images should be aligned with questions to extract answers with high precision.

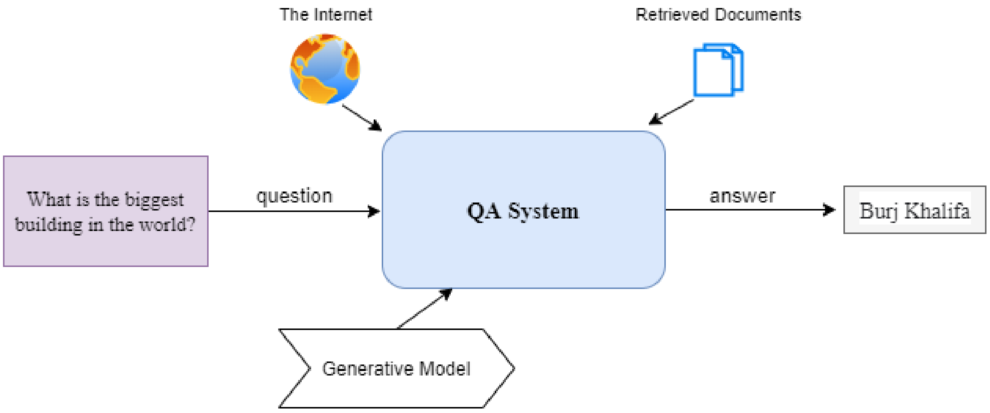


Figure 6.6 QA System

6.2.1 Question Answering in Natural Language Processing

Question Answering is an information retrieval task which extracts the answer from the context in Natural Language Processing. Mostly, the system finds the answer to the question in its knowledge. These kind of systems are called knowledge based systems. Beside that, there is also IR (Information Retrieval) based systems.

Question Answering systems can be classified as Extractive QA and Abstractive QA. Also there is such a distinction between the QA systems as open domain systems and closed domain systems which denotes the range of answer space.

Questions can be classified as yes/no questions, number questions and other questions like what, where, who or which. These questions can be evaluated with different metrics or methods. "Wh" questions can be evaluated with single or multiple answer. If the answer is in the first k answer, prediction can be assumed true but yes/no and number questions cannot be evaluated with this method. These concepts will be detailed in next sections with the implementation of Visual Question Answering.

6.2.2 Visual Question Answering (VQA)

Questions are not always asked to documents, also can be asked to pictures. In visual question answering, the system understands the question as it is in previous part but it also must understand the image and answer the question by looking at the picture.

Question types and answer types are the same with mentioned above. Answers can be found in answer space which is the list of all answers in the dataset. In this approach, the system can be called closed domain and extractive. Another approach is the generative approach which is using a generative language model to answer the question.

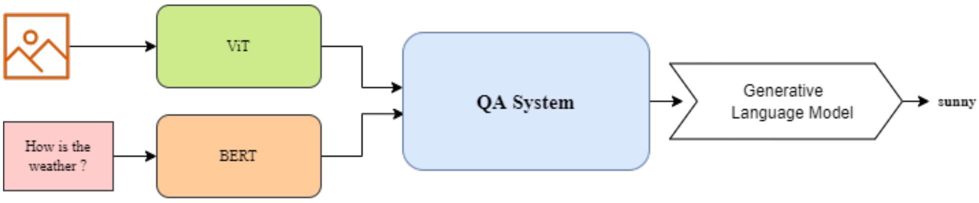


Figure 6.7 Generative Approach

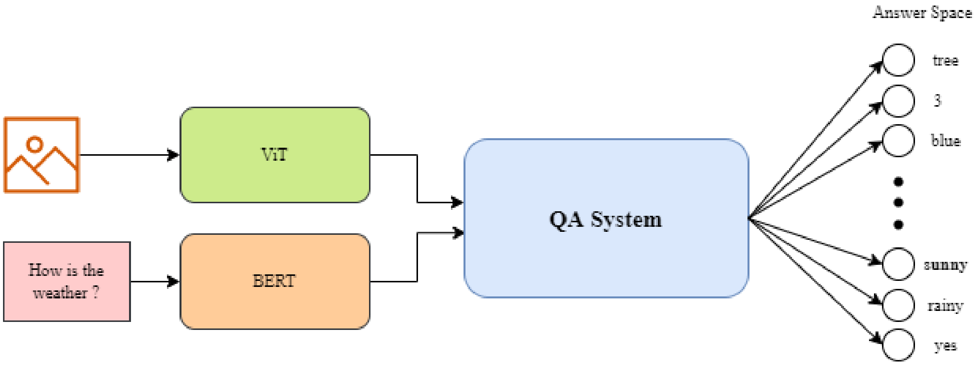


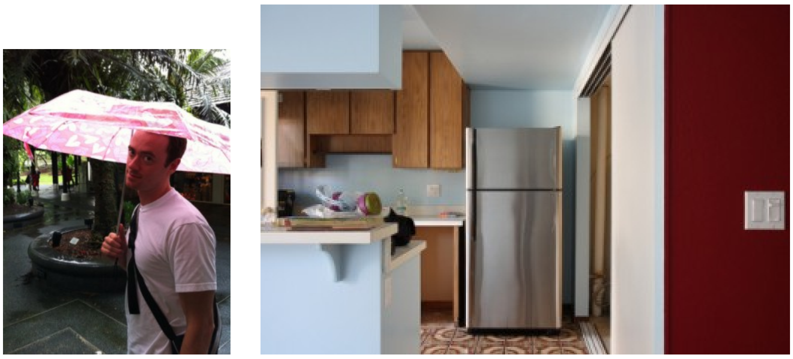
Figure 6.8 Answer Space Approach

We use answer space approach which is used for visual question answering task. Understanding of image with Visual Transformer (ViT) and understanding of question with BERT are explained in the image retrieval section. Structure for these two is the same with the previous task.

The interaction layer of this task is a fully connected network. Concatenation of `[CLS]` tokens of image and question is used for input to the fully connected layer. After one hidden layer sized 512 or 1024 which can be changed according to the size of the answer space, the output layer predict the answer by looking at the softmax results of the last layer which consists each unique answer in the dataset. Higher answer spaces need more complex fully connected networks. Beside that, bigger answer space means wider range of answers that can be predicted to the question. Used dataset is an important factor because of this constraint. Used datasets for this task will be detailed in next section.

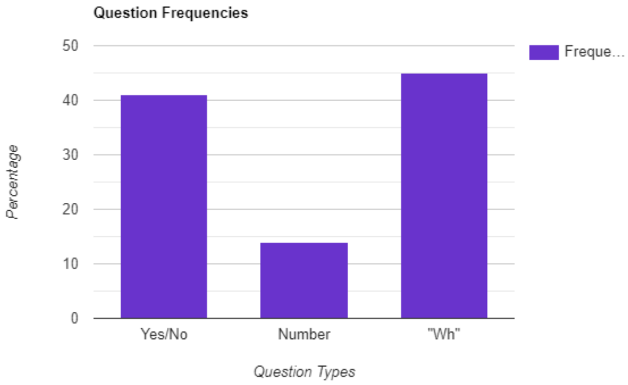
6.2.3 Benchmark Datasets in VQA Task

There are lots of image-question-answer pairs for using in this task in the literature. VQA: Visual Question Answering is the most used and most well-known dataset for Visual Question Answering.



(a) Question: What color is the umbrella? Answer: pink (Label: 1413)

(b) Question: What color is the refrigerator? Answer: silver (Label: 8282)



(c) Distribution of Questions

Figure 6.9 Sampled Examples and Distribution of Questions in VQA Dataset

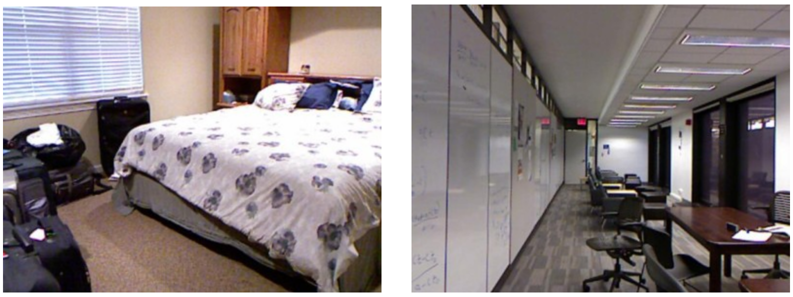
6.2.3.1 VQA Dataset

This dataset uses COCO Dataset's images for image database including 204k images and 760k questions with 10 million answers to these questions [48].

In the answer space approach, answers are made of one word because the model choose one of the answers in the answer space. This dataset has questions which have the answer that includes more than one word, so these questions and answers is removed from dataset before the training.

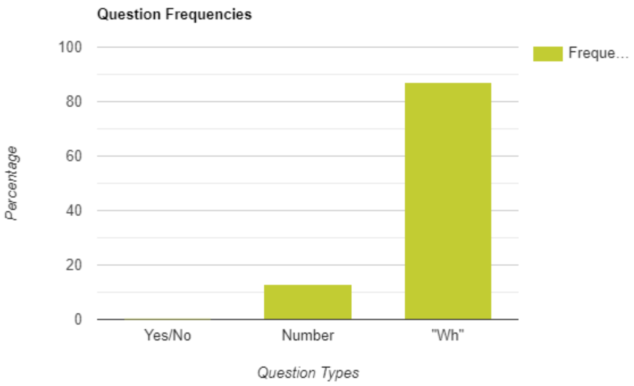
6.2.4 Metrics and Loss Functions in VQA Task

The loss function of the all trainings in the Visual Question Answering task is WuPalmer Similarity in this project. This metric calculates how much related the predicted answers and ground truth answers to each other. It looks the depths of the synsets in the WordNet taxonomies for the least common subsumer of prediction and ground truth words. Then using this formula, similarity is found.



(a) Question: How many beds are there?
Answer: 1 (Label: 0)

(b) Question: What is on the wall on the left side of the room?
Answer: whiteboard (Label: 565)



(c) Distribution of Questions

Figure 6.10 Sampled Examples and Distribution of Questions in DAQUAR Dataset

$$\text{sim}_{\text{wu-palmer}} = 2 \times \frac{\text{depth}(\text{lsc}(gt, pred))}{\text{depth}(gt) + \text{depth}(pred)} \tag{6.13}$$

The WuPalmer Similarity score is always bigger than 0 and smaller than or equal to 1. It cannot be smaller than zero because of the "Least Common Subsumer" function. This function never returns 0 because even the depth of the root of taxonomy is 1.

Similarity threshold is assigned as 0.925 for both datasets and the similarity of batches is found by mean operation of similarities of each prediction ground truth pairs.
Evaluation of the model is done by looking at the accuracy which calculates the exact matches, mean for the WuPalmer Similarity of all predictions and F1 score. F1 is a weak evaluation metric for this task, so it is not reported in this work. Accuracy is calculated according to the type of the questions beside its regular exact match calculation. Yes/No questions, number questions and "Wh" questions are considered separately in this task. Also for the "Wh" questions, number of first k answers that returned from model are also considered for k equals to 1, 3, 5 and 10.

7
RESULTS AND PERFORMANCE ANALYSIS

In this section, examples from inference time and performance of both VQA and Retrieval models are reported.

7.1 Image-Text Matching Model

To evaluate a image retrieval model, accuracy of mismatch prediction and top *k* recall (R@*k*) is calculated, where recall is defined as

$$R = \frac{\mid \{\text{Relevant Images}\} \cap \{\text{Retrieved Images}\} \mid}{\mid \{\text{Relevant Images}\} \mid} \tag{7.1}$$

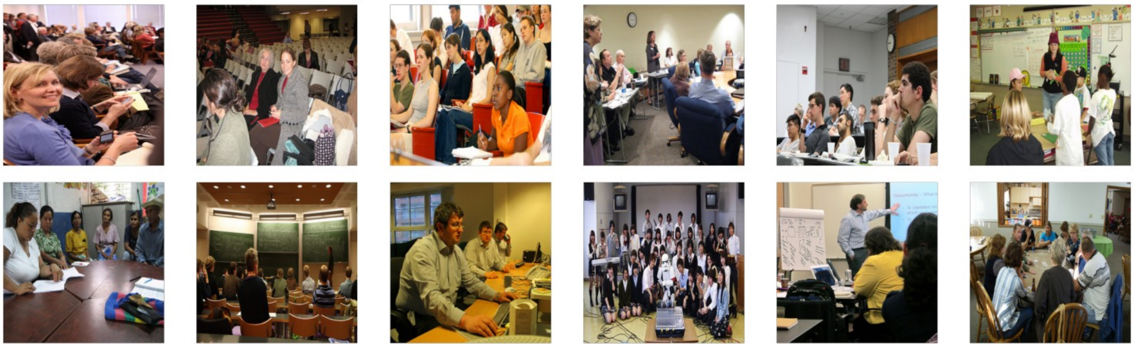
As shown in Table 7.1, model is achieved 20.2% for R@1, 56.2% for R@5, and 70.0% for R@10. This performance is expected due to each image has similar images in test and training sets. Hence, it will be more proper to look at R@10. Accuracy for match/mismatch objective is 87% at the end of third epoch.

In Figure 7.1 and Figure 7.2, we pass queries to our image database, which has nearly 4000 images, and retrieve 12 images for each query. It is shown that for straightforward and complex queries, our model is successful to retrieve relevant images. Admittedly, for query "the band is recording their new album which is going to be on markets in this year!!!" is not successful to retrieve relevant images. This is due to the database does not contain "recording album" images, however, it is successful to retrieve "bands". In the same way, for query "a beautiful view over blue sea, where kites are flowing at the air", retrieved images relevant. It can be seen that the model could not capture the "flowing kites" keyword, however, it captures "flowing fishing-net" or "flowing seagulls". At last, the query "we are witnessing an important historical moment" in Figure 7.2 can be classified as ambiguous, nevertheless, the model retrieved interesting images, which can be told that "important historical moments" intuitively.

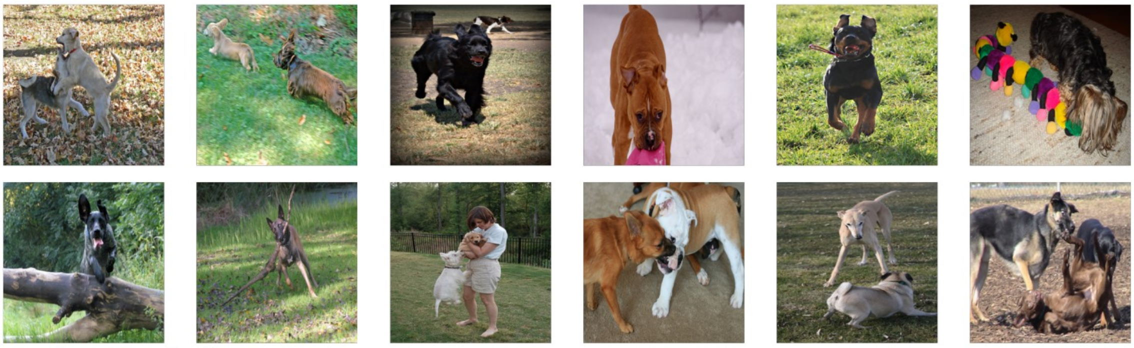
Table 7.1 Test Set Metrics for Flickr30k on Image-Text Matching Objective

Flickr30k (1K Test Set)			
Accuracy	**R@1**	**R@5**	**R@10**
87%	20.2	56.2	70

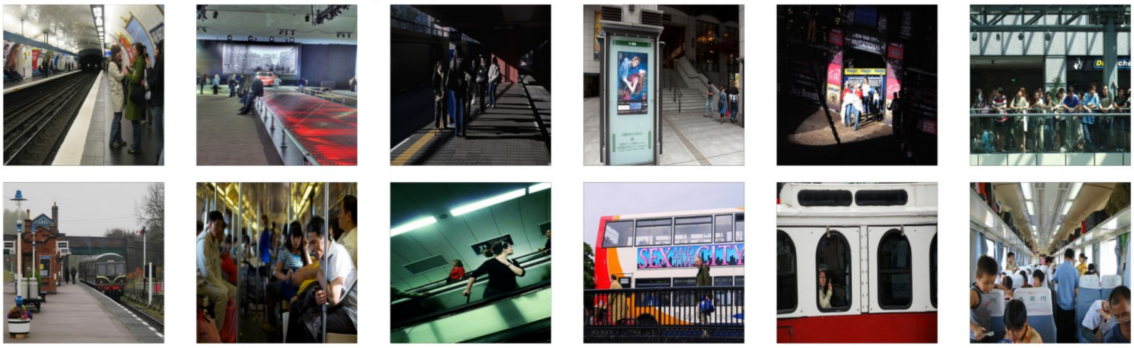
professor and student presenting their paper at conference at canada



two cute dogs



people waiting for the next subway



a software engineer is coding

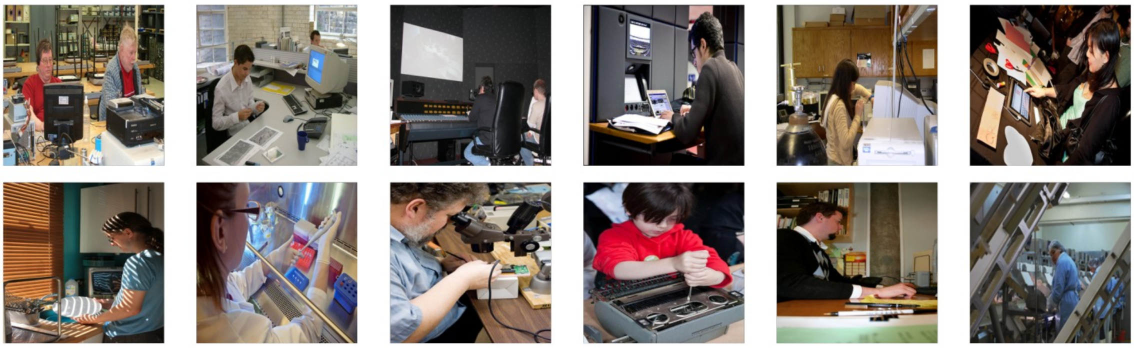
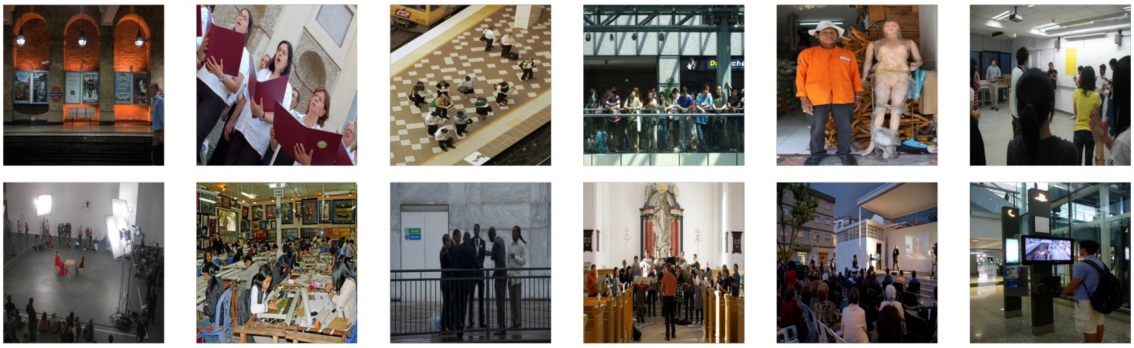
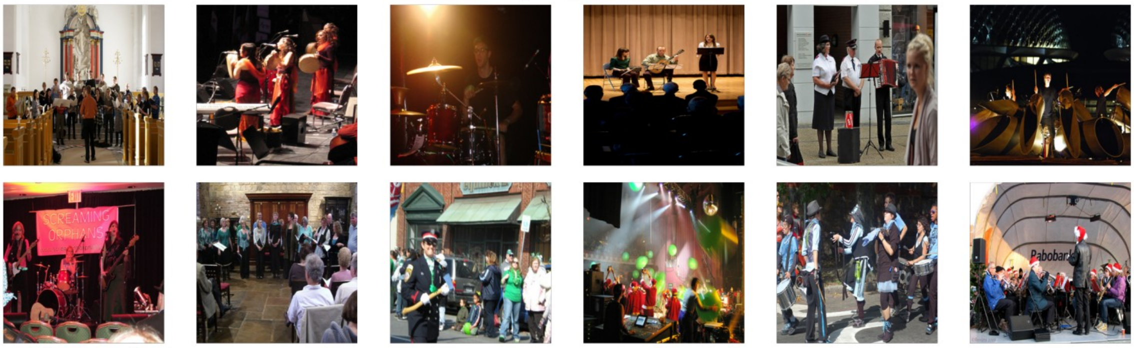


Figure 7.1 Twelve retrieved images for each four query. Images are stored and ranked from 4000 images.

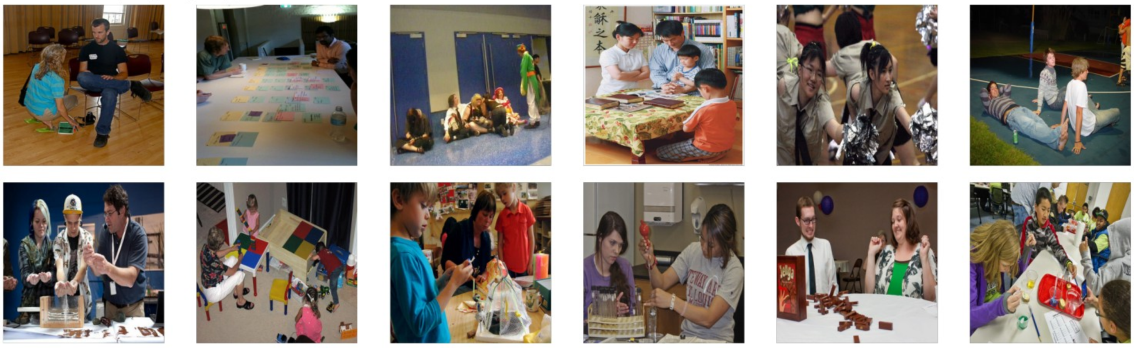
we are witnessing an important historical moment



the band is recording their new album which is going to on markets in this year!!!



adults are playing a game



a beautiful view over blue sea, where kites are flowing at the air

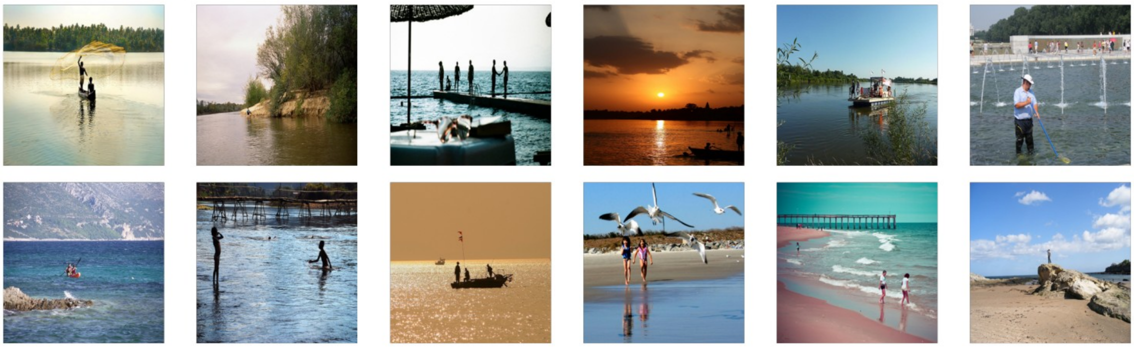


Figure 7.2 Twelve retrieved images for each four query. Images are stored and ranked from 4000 images.

7.2 Metric Learning Model

Results are shown in Table 7.2. Model is achieved 24.6% for R@1, 58.8% for R@5, and 76.7% for R@10. This performance is expected due to each image has similar

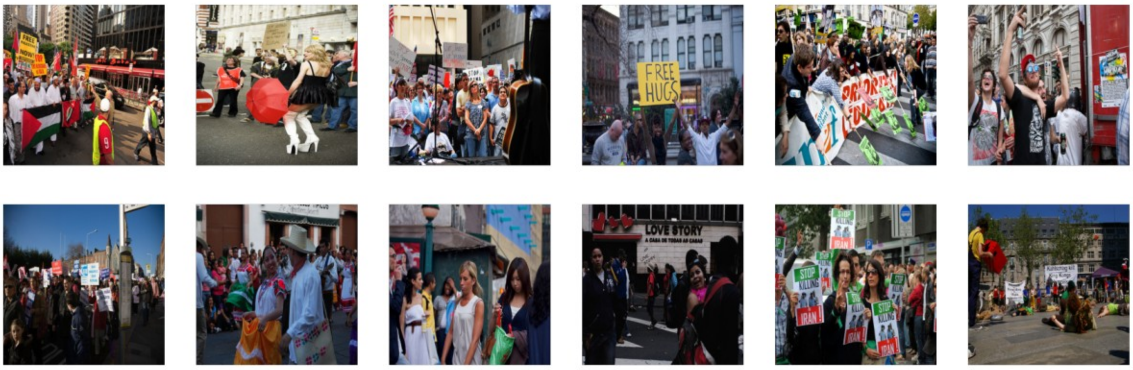
Table 7.2 Test Set Metrics for Flickr30k on Deep Metric Learning Objective

Flickr30k (1K Test Set)		
R@1	**R@5**	**R@10**
24.6	58.8	76.6

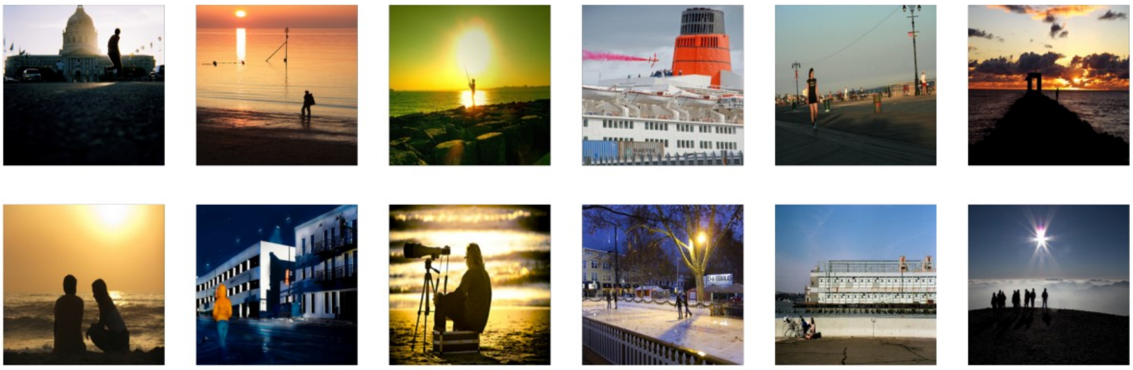
images in test and training sets. Hence, it will be more proper to look at R@10.

In Figure 7.3 and Figure 7.4, we pass queries to our image database, which has nearly 7000 images, and retrieve 12 images for each query. It is shown that for straightforward and complex queries, our model is successful to retrieve relevant images. CLIP like objective is more accurate and efficient than image-text matching. This is due to image-text matching uses a encoder layer for retrieval and do not use the power of joint space modeling. On the other hand, CLIP like model uses cosine similarity metric as an unbiased estimator. We pass more complex queries comparing to image-text matching model, such as "physician tries to split atom" gives very accurate and relevant images. For query "latest news from f1 racing", it's structure is unlikely comparing to others, nevertheless, it retrieves images that are relevant with "racing" and "formula one" cars.

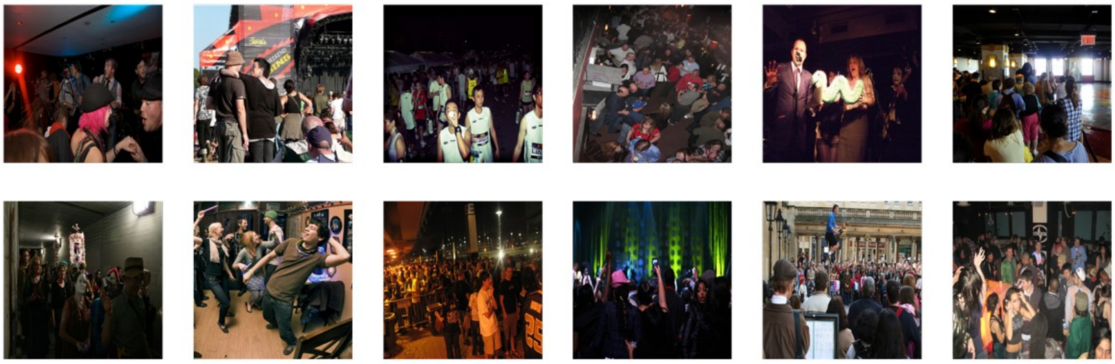
people are protesting something



a midnight view over sea



group of people having fun at the concert



students doing their exams

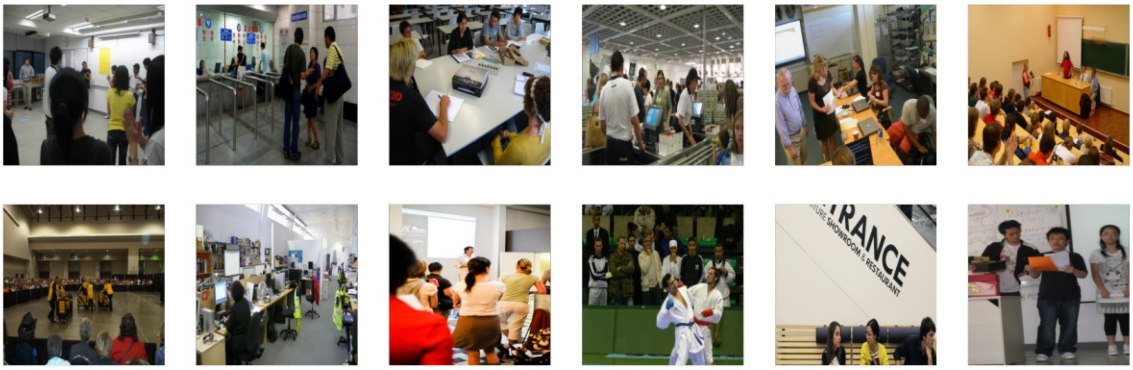
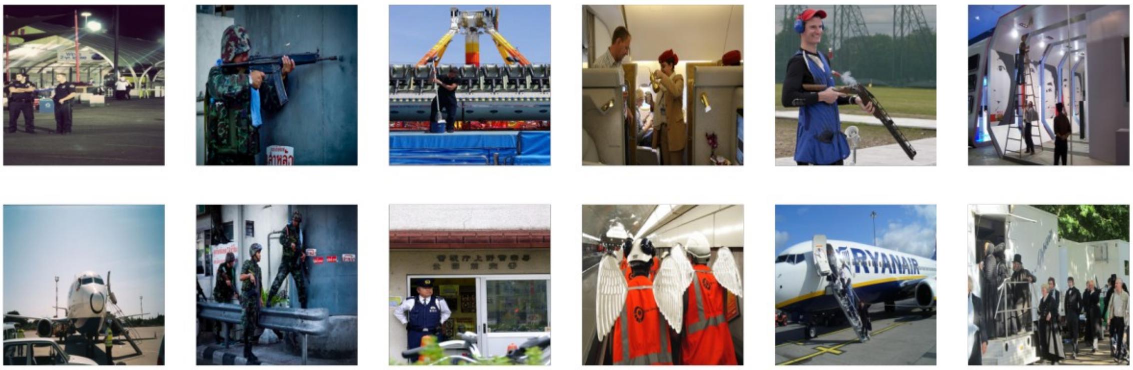
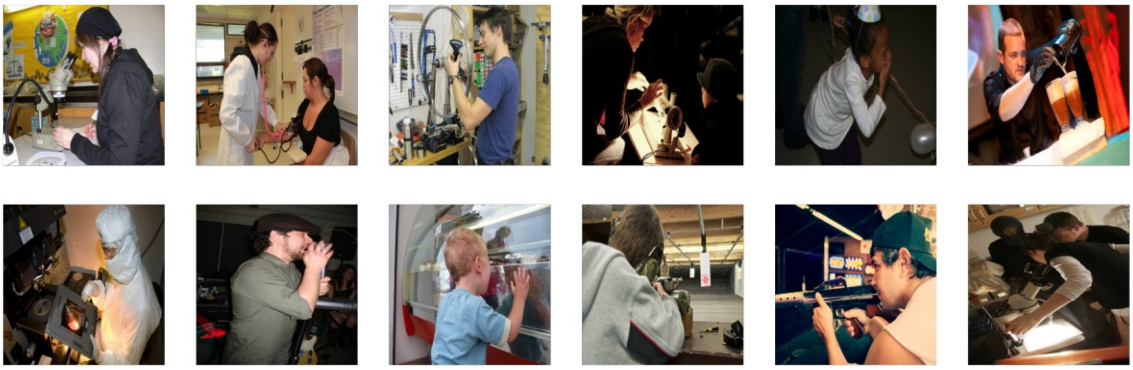


Figure 7.3 Twelve retrieved images for each four query. Images are stored and ranked from 7000 images.

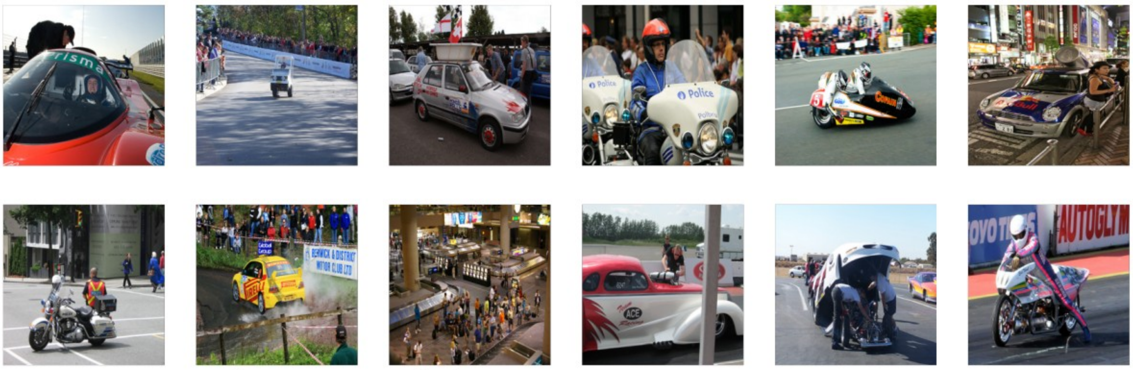
military is exercising



physician tries to split atom



latest news from f1 racing



fans support their team

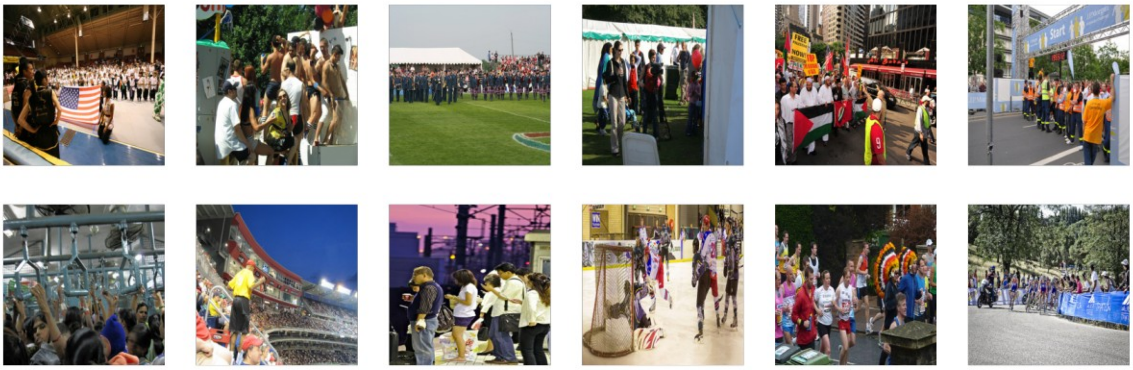


Figure 7.4 Twelve retrieved images for each four query. Images are stored and ranked from 7000 images.

Table 7.3 Different experimental results in percentage on VQA Dataset. @k indicates top-k accuracy.

Epochs	Loss	WuPS	Acc	Yes/No	Number	Wh@1	Wh@3	Wh@5	Wh@10)
2	2.37	46.48	42.99	58.68	28.16	33.3	52.2	59.6	66.78
3	2.32	48.08	44.67	58.5	30.65	36.45	55.03	61.89	69.17
4	2.324	49.07	45.61	58.85	30.04	37.64	56.6	63.58	70.57

7.3 VQA

There are 15 prediction results with the model that is trained with VQA Dataset for 4 epochs with the batch size of 64. Fully connected part of the networks has the size of 512. These examples are selected in validation dataset. Inference results are shown in Figure 6.15. With how many questions, the model cannot predict the exact number mostly, but if there are many of them, model also predict high numbers. Question concepts is understandable by model. This can be said by looking at "What is the color of baby's hair?", the model does not predict yellow, it predicts blonde. Also it can understand the references like this question "What are they doing?", question does not say about dogs but they word refer the dogs. The model can understand and answer the question correctly.

Evaluation is done with 20k sample with VQA Dataset. Hidden unit size is 512 for each experiment and batch size for evaluation is 64. Similarity threshold for Wu Palmer metric and loss function is set 0.925.

7.4 DAQUAR (DAtaset for QUestion Answering on Real-world images)

This is another dataset for VQA tasks which includes only one word answers. This dataset is more available for experimental works like changing the threshold or adding layer to the fully connected part of the network because it is smaller than the VQA dataset and its all answers are one word.

Also there is no yes/no question or "where" question in this dataset. Most of the questions include directions and locations with limited object information. So the comparison between the results of dataset made of other type of questions.

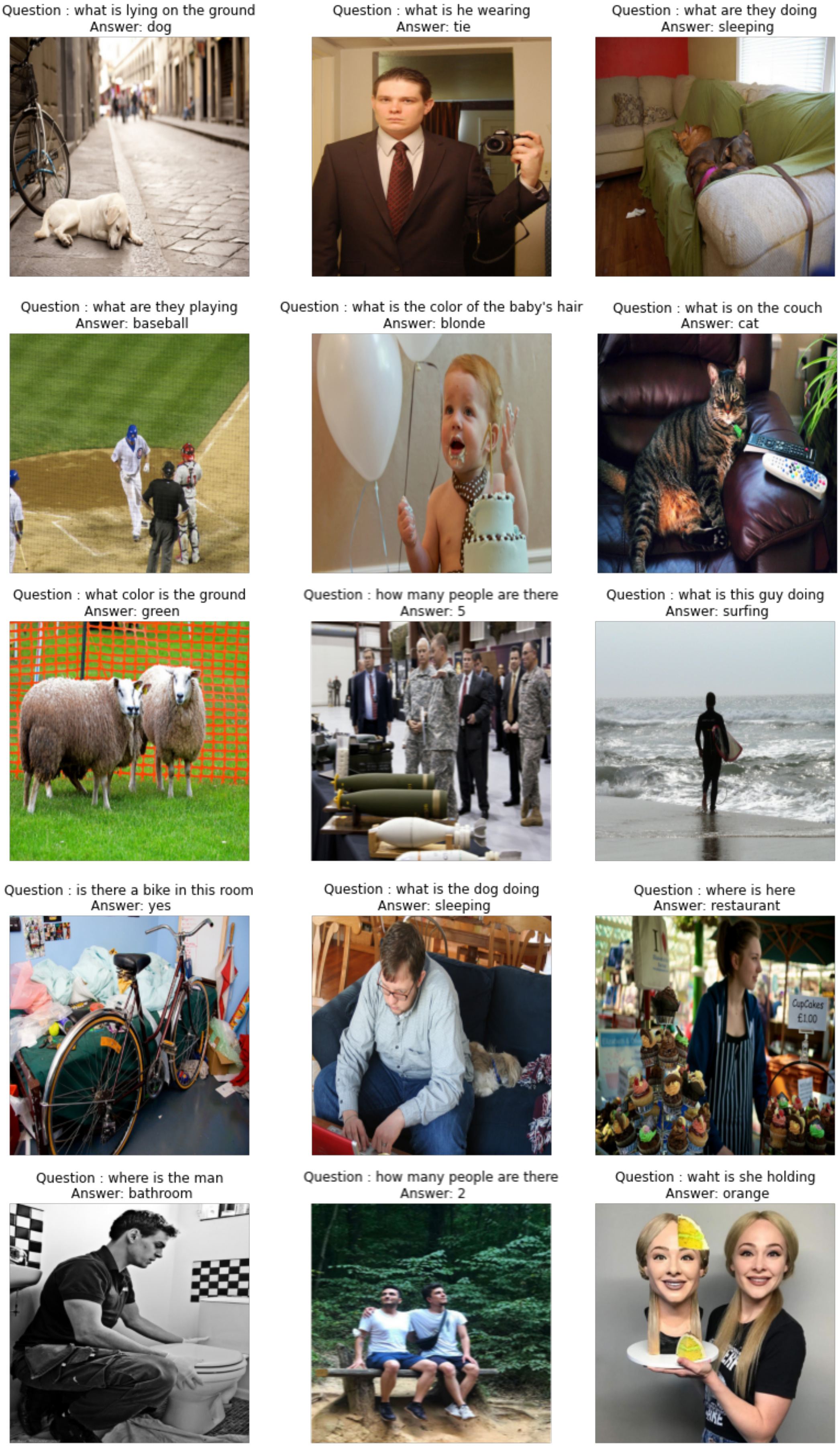
The Model is trained 30 epochs with the DAQUAR dataset with the batch size of 64. The interaction layer has 512 hidden units. Inference images are selected from validation set from the original dataset. Inference results are shown in Figure 6.16. This dataset is focused on directions, locations and several objects. This can be

Table 7.4 Different experimental results in percentage on DAQUAR Dataset. @k indicates top-k accuracy.

Epochs	Hidden Size	Loss	WuPS	Acc	Number	Wh@1	Wh@3	Wh@5	Wh@10
10	1024	3.701	28.84	23.55	35.58	17.48	31.51	39.78	50.09
30	512	3.873	31.51	26.42	28.31	26.11	42.21	50.11	60.54

understand by looking at images, almost all of them is indoor and bedroom or living room. Model cannot understand the "Where" question, that can be said by looking at the question "Where is this place?". This model trained on questions that includes directions or locations mostly. So this type of questions are easy to answer for model. Beside that there is no yes/no question in this dataset, so the model does not know this concept also.

Evaluation for DAQUAR dataset is done with all evaluation set of the original dataset that has 5673 samples with the batch size of 64. Similarity threshold for Wu Palmer metric and loss function is set 0.925.



Question : what is lying on the ground
Answer: dog

Question : what is he wearing
Answer: tie

Question : what are they doing
Answer: sleeping

Question : what are they playing
Answer: baseball

Question : what is the color of the baby's hair
Answer: blonde

Question : what is on the couch
Answer: cat

Question : what color is the ground
Answer: green

Question : how many people are there
Answer: 5

Question : what is this guy doing
Answer: surfing

Question : is there a bike in this room
Answer: yes

Question : what is the dog doing
Answer: sleeping

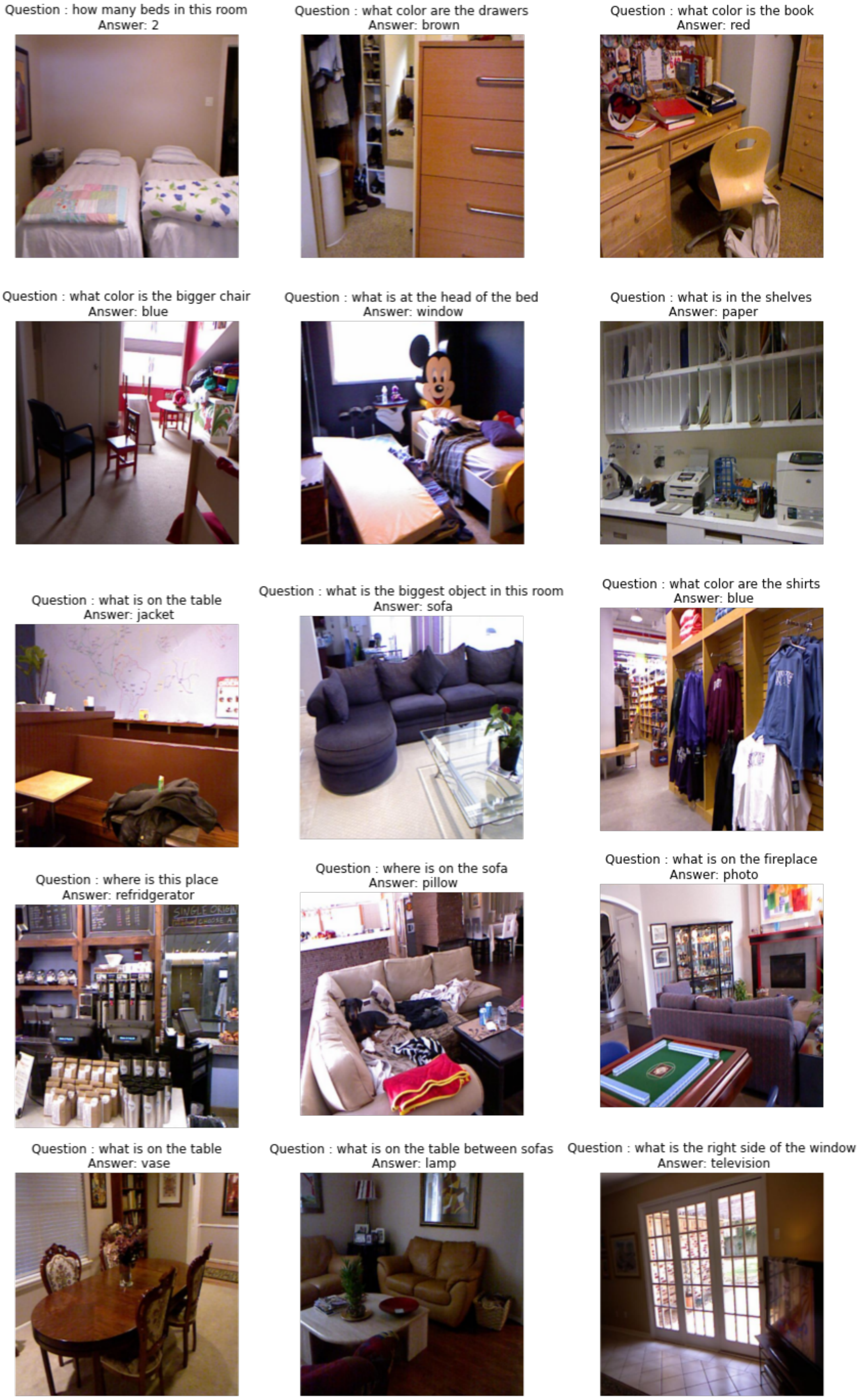
Question : where is here
Answer: restaurant

Question : where is the man
Answer: bathroom

Question : how many people are there
Answer: 2

Question : waht is she holding
Answer: orange

Figure 7.5 Model's Prediction Examples for VQA



Question : how many beds in this room
Answer: 2

Question : what color are the drawers
Answer: brown

Question : what color is the book
Answer: red

Question : what color is the bigger chair
Answer: blue

Question : what is at the head of the bed
Answer: window

Question : what is in the shelves
Answer: paper

Question : what is on the table
Answer: jacket

Question : what is the biggest object in this room
Answer: sofa

Question : what color are the shirts
Answer: blue

Question : where is this place
Answer: refridgerator

Question : where is on the sofa
Answer: pillow

Question : what is on the fireplace
Answer: photo

Question : what is on the table
Answer: vase

Question : what is on the table between sofas
Answer: lamp

Question : what is the right side of the window
Answer: television

Figure 7.6 Model's Prediction Examples for DAQUAR

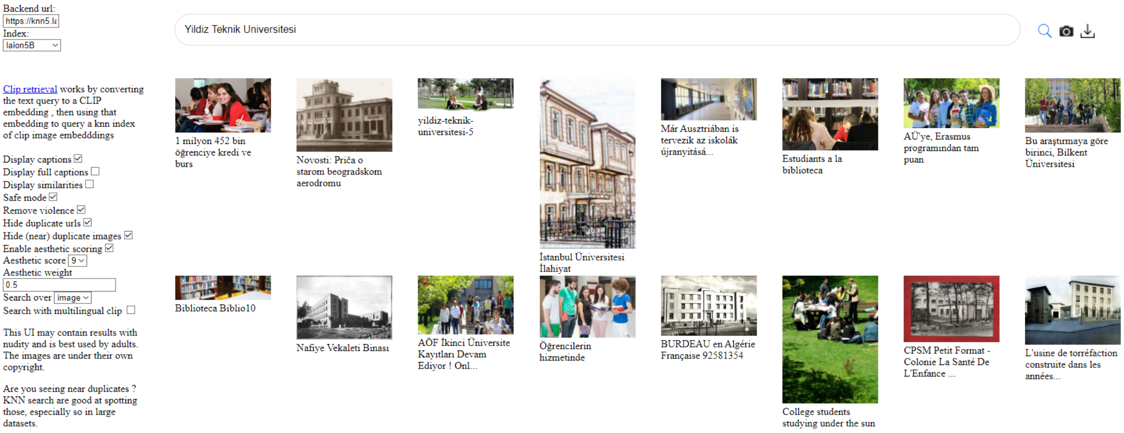
8

TIR: A NEW BENCHMARK FOR "T"URKISH "I"MAGE "R"ETRIEVAL AT SCALE

In this thesis, our novel contributions are mainly focused on introducing a new image retrieval benchmark, entitled TIR (**T**urkish **I**mage **R**etrieval), and a fine-tuned minimalist CLIP model on this benchmark.

In April 2022, an Image Retrieval dataset called LAION-5B (A NEW ERA OF OPEN LARGE-SCALE MULTI-MODAL DATASETS) [49] is released. This dataset contains 5.85 billion Image-Text pairs which are open source completely. It contains 2 billion of English pairs, 2 billion of multilingual pairs and 1 billion with no language. Multilingual part of this dataset contains also Turkish samples. LAION-5B benchmark is served to strengthen and make open the image-text models. So, the motivation of using LAION-5B benchmark is to contribute to the literature a perfectly compiled monolingual (Turkish) text2image retrieval dataset for the first time.

The LAION-5B benchmark has size of 800GB, with pairs of image URL and its caption. The license of the benchmark is Creative Common CC-BY 4.0 license, which means there is no particular restriction. Due to the high amount size, we are not able to download whole dataset. Hence, we perform lazy evaluation to get *n* samples from



Backend url:
https://knn5.la
Index:
laion5B

Yildiz Teknik Universitesi

Clip retrieval works by converting the text query to a CLIP embedding , then using that embedding to query a knn index of clip image embedddings

Display captions ☑
Display full captions ☐
Display similarities ☐
Safe mode ☑
Remove violence ☑
Hide duplicate urls ☑
Hide (near) duplicate images ☑
Enable aesthetic scoring ☑
Aesthetic score 9
Aesthetic weight
0.5
Search over image
Search with multilingual clip ☐

This UI may contain results with nudity and is best used by adults. The images are under their own copyright.

Are you seeing near duplicates ? KNN search are good at spotting those, especially so in large datasets.

1 milyon 452 bin öğrenciye kredi ve burs
Novosti: Priča o starom beogradskom aerodromu
yildiz-teknik-universitesi-5
İstanbul Üniversitesi İlahiyat
Már Ausztriában is tervezik az iskolák újranyitásá...
Estudiants a la biblioteca
AÜ'ye, Erasmus programından tam puan
Bu araştırmaya göre birinci, Bilkent Üniversitesi
Biblioteca Biblio10
Nafiye Vekaleti Binası
AÖF İkinci Üniversite Kayıtları Devam Ediyor ! Onl...
Öğrencilerin hizmetinde
BURDEAU en Algérie Française 92581354
College students studying under the sun
CPSM Petit Format - Colonie La Santé De L'Enfance ...
L'usine de torréfaction construite dans les années...

Figure 8.1 LAION Dataset

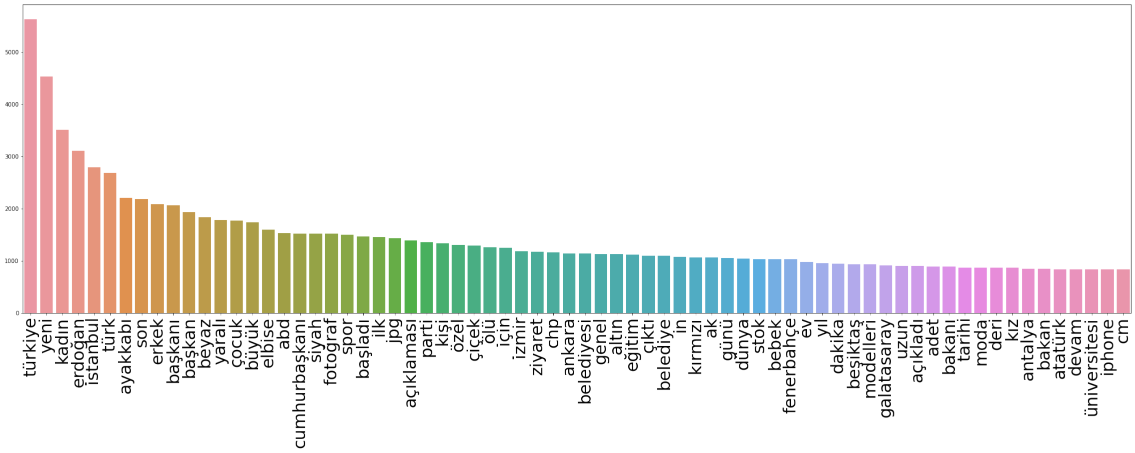


Figure 8.2 Most frequent 70 words from TIR dataset

the dataset, which means we download samples when it is required. We shuffle the dataset with different seeds to get different samples for lazy evaluation.

169000 samples are collected from multilingual version of the LAION-5B by looking their language with a simple filtering method. Besides, we sample 182000 English image-text pairs. Since the dataset stores images as URL, it is needed to crawl this images. We write a simple image crawler and run this script on two machines at parallel for one week.

The motivation of English samples is that Turkish images reflects the cultural values of Turkish language, however, we also want to enrich the sentence diversity in our benchmark and unbias the distribution of both images and texts. Thus, we translate the English captions with a pre-trained Neural Machine Translation model. Captions of images are translated with Opus-MT [50] model. Opus-MT is a BART based pre-trained language model. After this process, Turkish Image-Text pairs and translated Turkish Image-Text pairs are combined for the training of minimalist CLIP model.

8.1 Dataset Characteristics

Before training a image retrieval model, captions are pre-processed with function in Algorithm 3.

After pre-processing, the total number of the tokens is 159225 and vocabulary size (number of unique tokens) $|V|$ is 142982. To illustrate the dataset word distribution, most frequent 70 words from TIR dataset is visualized in Figure 8.2. For visualization, we remove numbers and stop words in Turkish.

Table 8.1 Query Translations/Literal Translations for TIR Image Retrieval Examples

F - E	Turkish Query	Translation / Literal Translation
8.3 - 1	enflasyon oranını duyan halk çıldırdı	hearing the inflation rate, the people went crazy
8.3 - 2	çeyrek altın	quarter gold
8.3 - 3	koronavirus salgını ülke geneldinde pandemiyi devam ettiriyor	coronavirus epidemic continues the pandemic across the country
8.3 - 4	kedi ve köpek	cat and dog
8.4 - 1	istanbudaki çocuk eğlence merkezleri	children’s entertainment centers in istanbul
8.4 - 2	cuma namazı nasıl kılınır	how to friday pray
8.4 - 3	yıldız teknik üniversitesi	yildiz technical university
8.4 - 4	galatasaray şampiyon	galatasaray is the champion
8.5 - 1	sınava hazırlanan öğrenciler	students preparing for the exams
8.5 - 2	yeni film sinemalarda	new movie is at the cinemas
8.5 - 3	cimbom şampiyon	cimbom is the champion
8.5 - 4	barajlardaki su oranı azalıyor	water content in dams is decreasing
8.6 - 1	bilgisayar oyunu	computer game
8.6 - 2	seçim sonuçları açıklanmaya devam ediyor	election results continue to be announced
8.6 - 3	yemek tarifi	recipe
8.6 - 4	yaz sıcaklarıyla insanlar sahillere ve plajlara doldu	with the summer heat, people filled the beaches and plages.

Table 8.2 Image Resolution Statistics of TIR.

	Mean	**Max**	**Min**	**Std**
Width	374.59	23937	10	284.17
Height	518.42	8287	16	329.80

All images are stored as RGB images. Mean width of images is 374.59 and mean height of images is 518.42 (other statistics are shown in Table 8.1). This resolutions are reduced to (3 × 224 × 224) during training. To normalize the pixel distribution of images, we use the formula in Equation 8.1.

$$I = \frac{I - \mu_I * \max_I}{\sigma_I * \max_I} \tag{8.1}$$

The 80% portion of dataset is used for training and 20% for testing. Test samples are obtained from the last 20% samples of released dataset.

Algorithm 3 pre-process

```
Require: dataset $D$, a sample text $S$
$S$ = re.sub(r"’[a-za-z0-9ğüşöçıiğüşöç]+[ ]*", ’ ’, $S$).rstrip()
$S$ = re.sub(r’\[[^]]*\]’, ’ ’, $S$)
$S$ = re.sub(r’[^a-zA-Z\n \w0-9 ]’, ’ ’, $S$)
$S$ = re.sub(r’[ ]2,’, ’ ’, $S$)
$S$ = $S$.lower().rstrip()
return $S$
```

8.2 Results

We train a minimalist implementation of CLIP model, which is described in Section 6.1.2 in detail, with the exact same parameters of model introduced in Section 6.1.2. We follow the same training hyperparameters and inference algorithms. 16 different queries and retrieved images are shown in Figure 8.3, 8.4, 8.5 and 8.6. The translations (or literal translations) are presented in Table 8.1 for non-Turkish speakers.

As can be seen in below figures, our retrieval model works fine when it comes to simple and complex queries. For example, example 4 in Figure 8.3, relevant images are retrieved from a basic query "kedi ve kopek". Our system also reflects the cultural features, for example example 2 in Figure 8.6 retrieves images from part elections and exam results.

When it comes to advanced level comprehension and understanding, example 4 in Figure 8.5 shows that the model retrieves "dry" barrage, however the query does not have "dry". It has a semantic approximated phrase "su orani azaliyor" (water rate is decreasing).

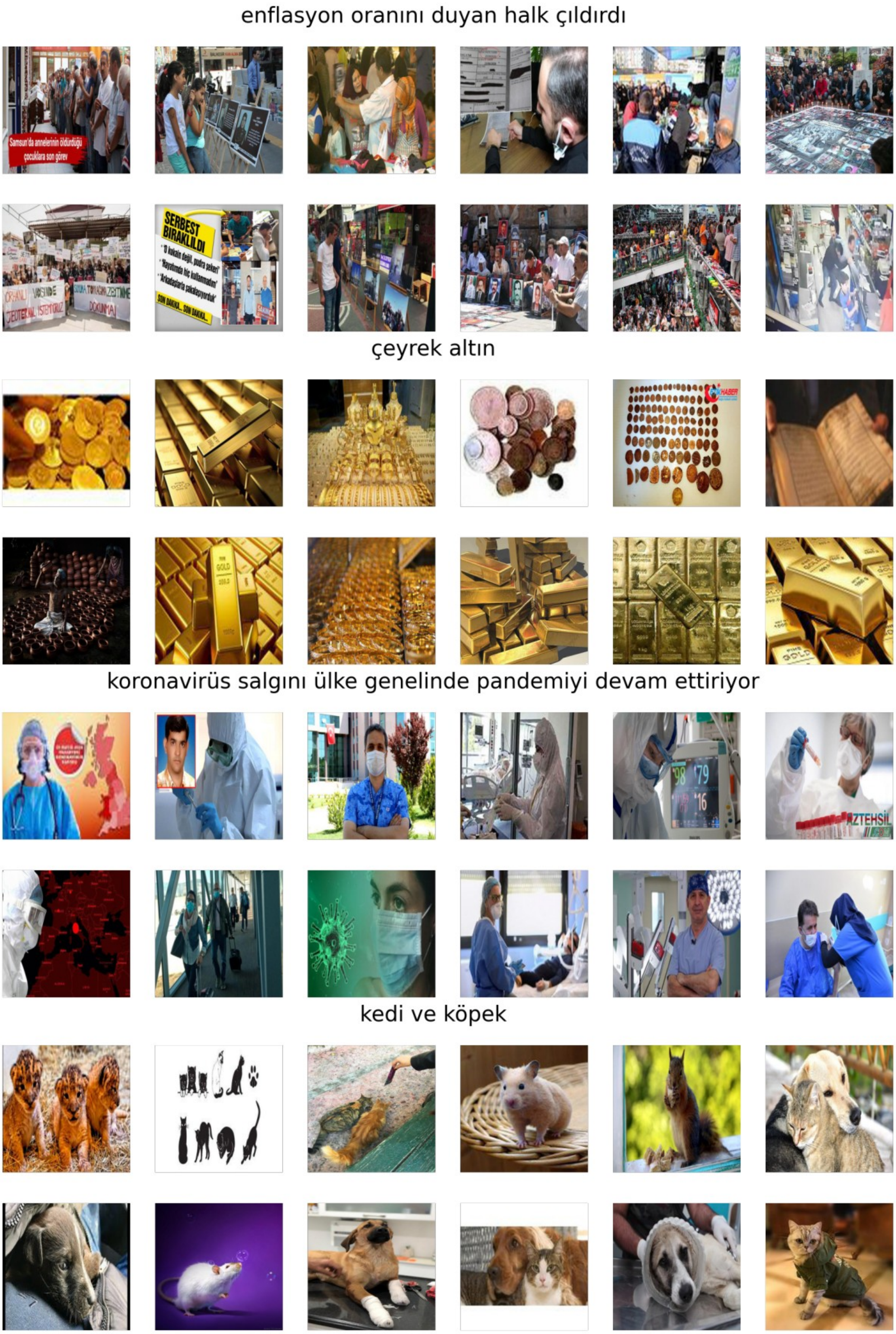

enflasyon oranını duyan halk çıldırdı
çeyrek altın
koronavirüs salgını ülke genelinde pandemiyi devam ettiriyor
kedi ve köpek

Figure 8.3 TIR Image Retrieval Examples (1): Twelve retrieved images for each four query. Images are stored and ranked from 150000 images.

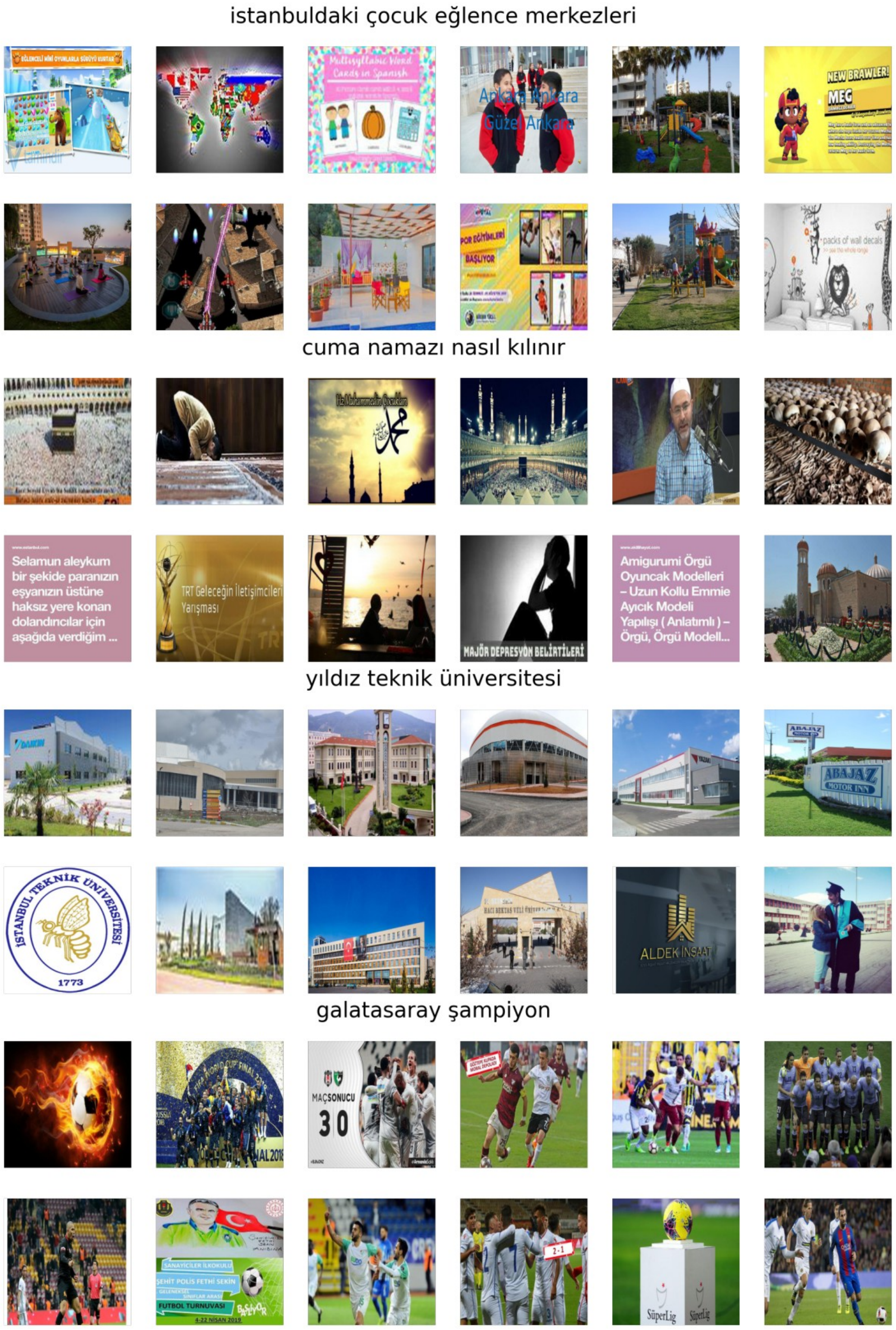

istanbuldaki çocuk eğlence merkezleri
cuma namazı nasıl kılınır
yıldız teknik üniversitesi
galatasaray şampiyon

Figure 8.4 TIR Image Retrieval Examples (2): Twelve retrieved images for each four query. Images are stored and ranked from 150000 images.

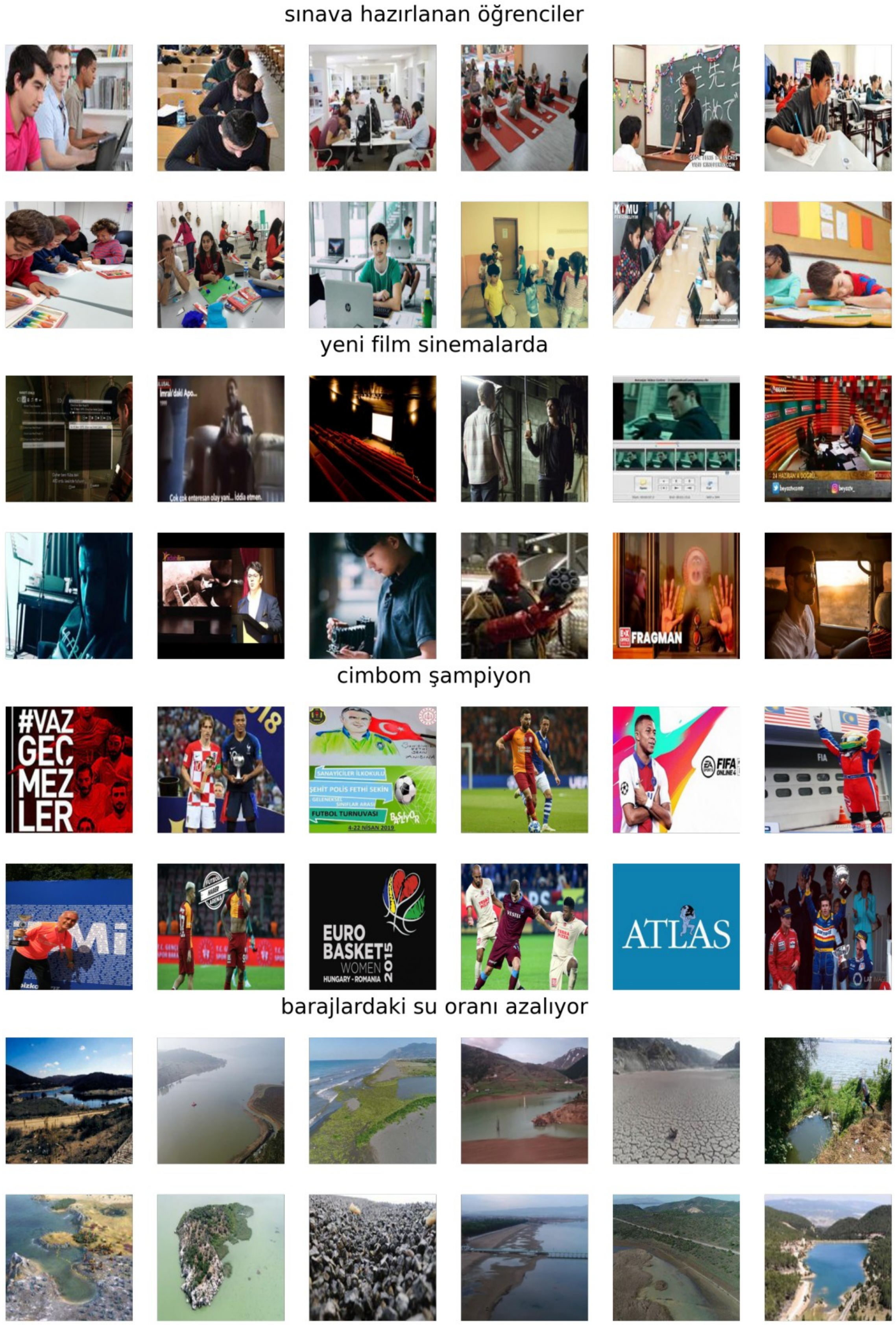

sınava hazırlanan öğrenciler
yeni film sinemalarda
cimbom şampiyon
barajlardaki su oranı azalıyor
#VAZ GEÇ MEZ LER
FRAGMAN
EURO BASKET 2015 WOMEN HUNGARY - ROMANIA
ATLAS

Figure 8.5 TIR Image Retrieval Examples (3): Twelve retrieved images for each four query. Images are stored and ranked from 150000 images.

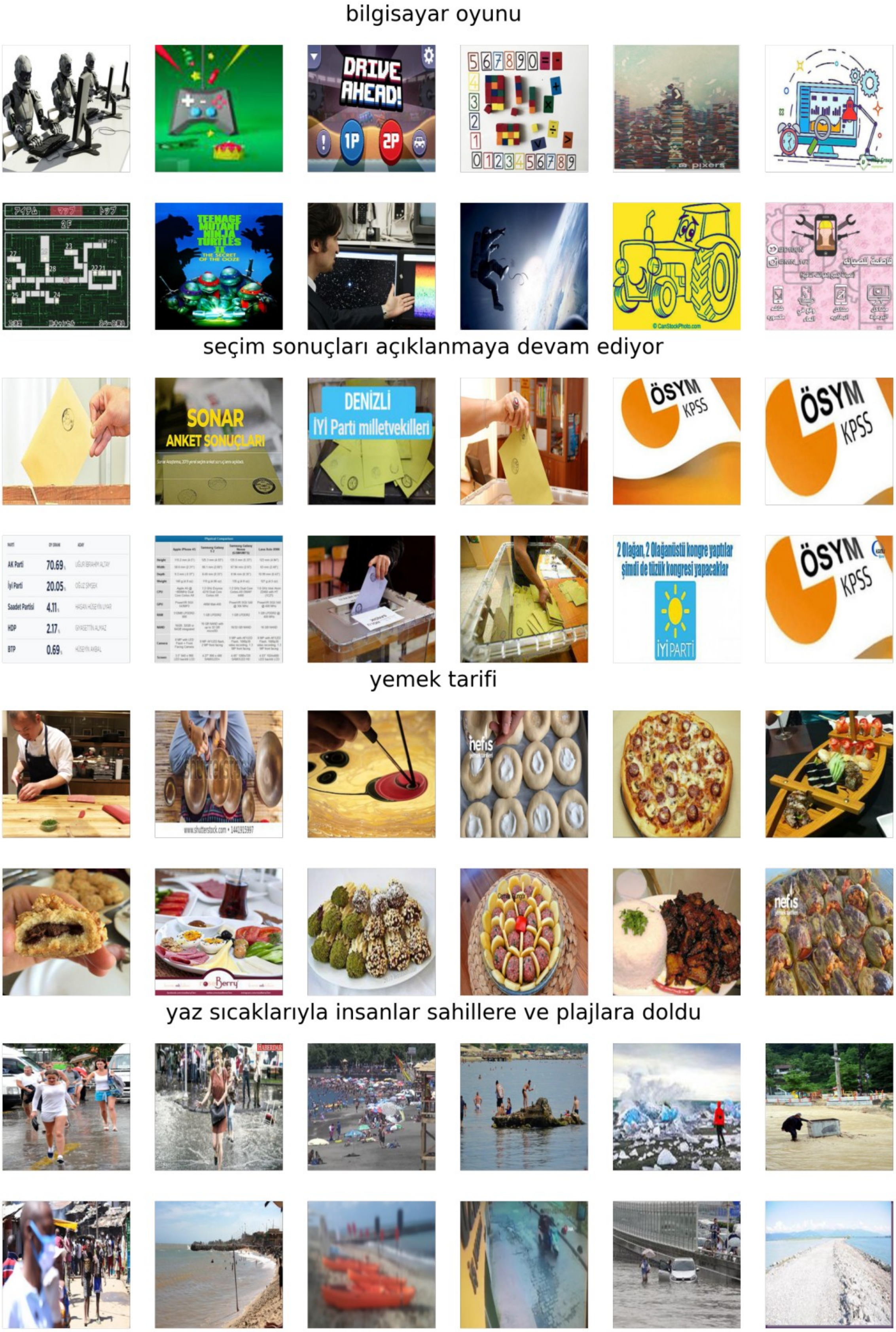

bilgisayar oyunu
seçim sonuçları açıklanmaya devam ediyor
yemek tarifi
yaz sıcaklarıyla insanlar sahillere ve plajlara doldu

Figure 8.6 TIR Image Retrieval Examples (4): Twelve retrieved images for each four query. Images are stored and ranked from 150000 images.

9

APPLICATIONS

9.1 A Vector Search Engine for Images

To illustrate our work, we create a visual semantic vector based search engine. A user pass the query to the system, then the search engine ranks and retrieves relevant images and serves to the user. At backend, we use trained minimal CLIP model. The application support both English and Turkish languages. There are two pre-located directories which act a role as image database. The vector representation of images in this directories are pre-calculated and mapped to relevant docids.

Besides that, if a user wants to search over him/her/their own database, new directory database must be named as "tr_custom" or "en_custom". For the first query, the system calculates all vectors in this database and maps relevant docids to each image. After first query, user will be able to retrieve images by search engine without re-calculation of vectors and docids.

The frontent of the application can be seen in Figure 9.1.

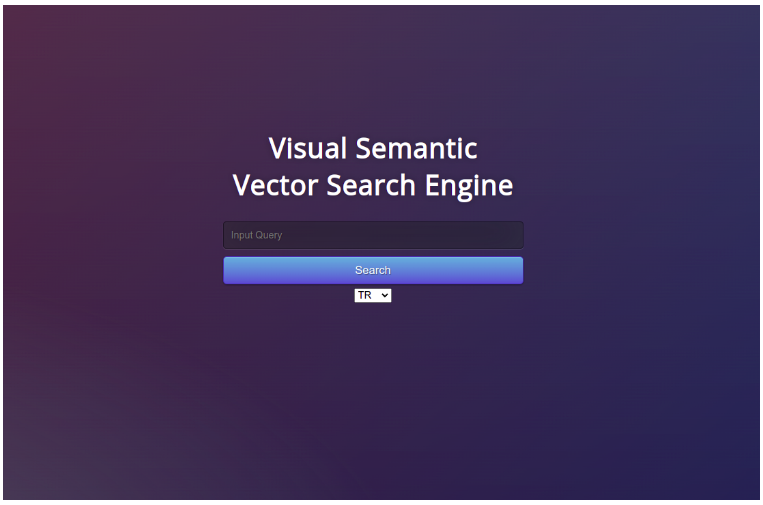


Figure 9.1 Demo Website for Retrieval Model

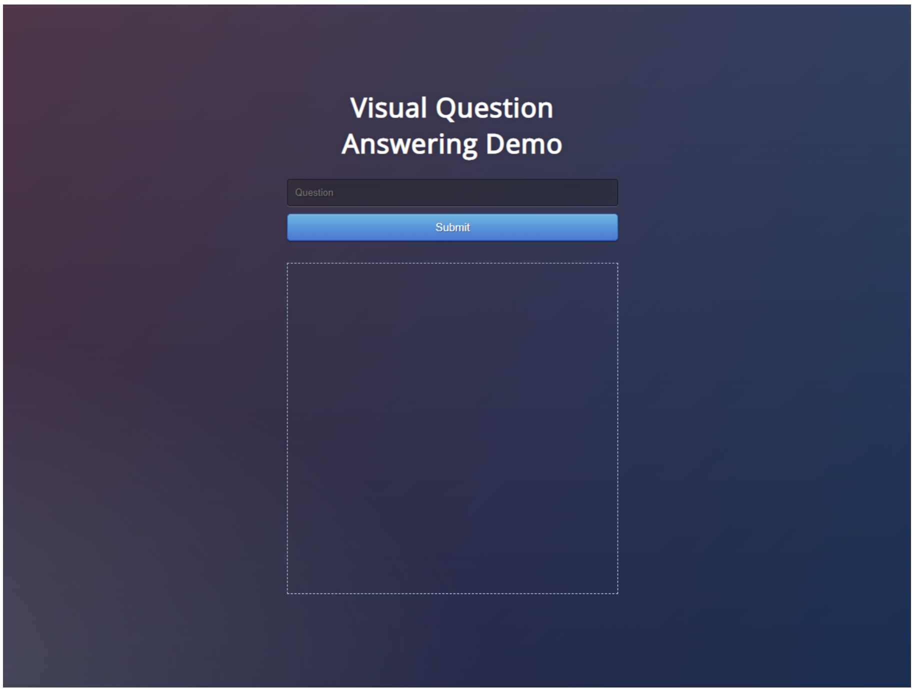


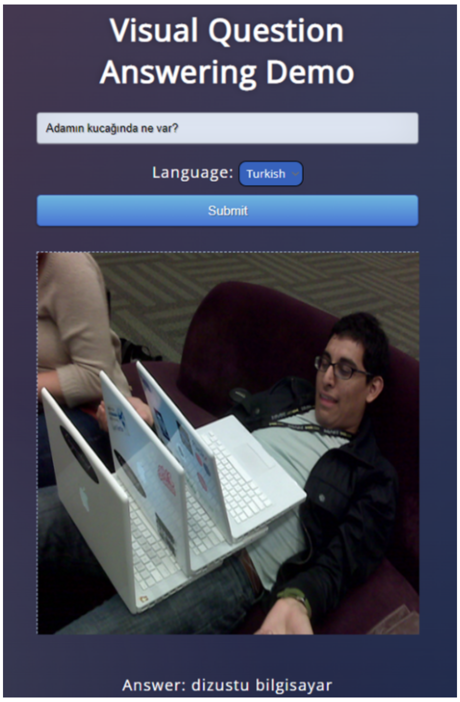
Figure 9.2 Demo Website for VQA Model

9.2 Multilingual Visual Question Answering Tool

Visual Question Answering part of this thesis is dressed up with a demo website. This website runs VQA Model which trained on COCO VQA Dataset, on the backend. On the frontend, there is a drag and drop area for uploading an image for asking questions about it. Above the drag and drop editor, there is a text input box for the question. When the question and image are given, it can be pressed to the submit button to see the predicted answer. Predicted answer can be seen below the uploaded image after prediction process is done.

Demo has also supports Turkish questions. If "TR" option is selected, answer will also be in Turkish. For Turkish, different model from the one in English is working on the backend. This model is trained on translated VQA COCO Model, so it learned all concepts in Turkish. Translation operation is done with the machine translation model which also used for translation of English LAION Samples. For text encoding transformer, Turkish Bert Model is used in this training.

Results that are taken from website can be seen in figures below this section.

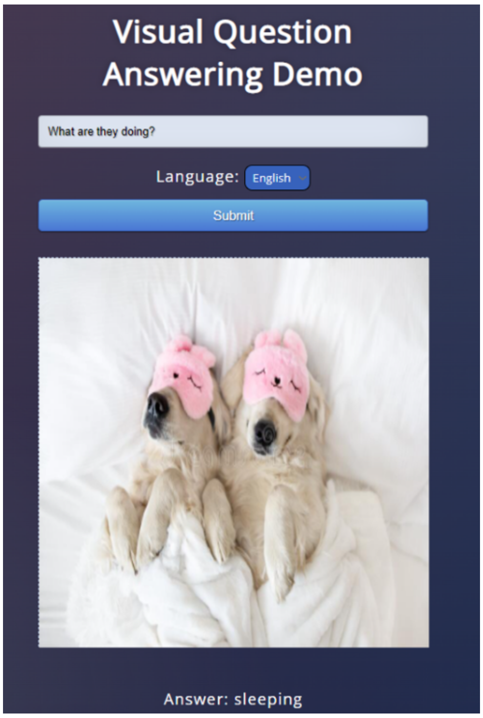


Adamın kucağında ne var?

Language: Turkish

Answer: dizustu bilgisayar

Figure 9.3 Turkish Output



What are they doing?

Language: English

Answer: sleeping

Figure 9.4 English Output

References

[1] R. C. Berwick and N. Chomsky, *Why Only Us: Language and Evolution*. The MIT Press, 2015, ISBN: 0262034247.

[2] T. Ates *et al.*, *Craft: A benchmark for causal reasoning about forces and interactions*, 2020. DOI: `10.48550/ARXIV.2012.04293`. [Online]. Available: `https://arxiv.org/abs/2012.04293`.

[3] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, *Natural language processing (almost) from scratch*, 2011. DOI: `10.48550/ARXIV.1103.0398`. [Online]. Available: `https://arxiv.org/abs/1103.0398`.

[4] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2014. DOI: `10.48550/ARXIV.1409.1556`. [Online]. Available: `https://arxiv.org/abs/1409.1556`.

[5] A. v. d. Oord *et al.*, *Wavenet: A generative model for raw audio*, 2016. DOI: `10.48550/ARXIV.1609.03499`. [Online]. Available: `https://arxiv.org/abs/1609.03499`.

[6] B. Rim, N.-J. Sung, S. Min, and M. Hong, "Deep learning in physiological signal data: A survey," *Sensors*, vol. 20, no. 4, 2020, ISSN: 1424-8220. DOI: `10.3390/s20040969`. [Online]. Available: `https://www.mdpi.com/1424-8220/20/4/969`.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. DOI: `10.1109/CVPR.2009.5206848`.

[8] S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks*, 2015. DOI: `10.48550/ARXIV.1506.01497`. [Online]. Available: `https://arxiv.org/abs/1506.01497`.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2018. DOI: `10.48550/ARXIV.1810.04805`. [Online]. Available: `https://arxiv.org/abs/1810.04805`.

[10] H. Zhou *et al.*, *Informer: Beyond efficient transformer for long sequence time-series forecasting*, 2020. DOI: `10.48550/ARXIV.2012.07436`. [Online]. Available: `https://arxiv.org/abs/2012.07436`.

[11] A. Vaswani *et al.*, *Attention is all you need*, 2017. DOI: `10.48550/ARXIV.1706.03762`. [Online]. Available: `https://arxiv.org/abs/1706.03762`.

[12] A. Radford *et al.*, *Learning transferable visual models from natural language supervision*, 2021. DOI: `10.48550/ARXIV.2103.00020`. [Online]. Available: `https://arxiv.org/abs/2103.00020`.

[13] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-rnn)," 2014. DOI: `10.48550/ARXIV.1412.6632`. [Online]. Available: `https://arxiv.org/abs/1412.6632`.

[14] J. Donahue *et al.*, *Decaf: A deep convolutional activation feature for generic visual recognition*, 2013. DOI: `10.48550/ARXIV.1310.1531`. [Online]. Available: `https://arxiv.org/abs/1310.1531`.

[15] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, *On using very large target vocabulary for neural machine translation*, 2014. DOI: `10.48550/ARXIV.1412.2007`. [Online]. Available: `https://arxiv.org/abs/1412.2007`.

[16] D. Bahdanau, K. Cho, and Y. Bengio, *Neural machine translation by jointly learning to align and translate*, 2014. DOI: `10.48550/ARXIV.1409.0473`. [Online]. Available: `https://arxiv.org/abs/1409.0473`.

[17] M.-T. Luong, H. Pham, and C. D. Manning, *Effective approaches to attention-based neural machine translation*, 2015. DOI: `10.48550/ARXIV.1508.04025`. [Online]. Available: `https://arxiv.org/abs/1508.04025`.

[18] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, *Recurrent models of visual attention*, 2014. DOI: `10.48550/ARXIV.1406.6247`. [Online]. Available: `https://arxiv.org/abs/1406.6247`.

[19] J. Ba, V. Mnih, and K. Kavukcuoglu, *Multiple object recognition with visual attention*, 2014. DOI: `10.48550/ARXIV.1412.7755`. [Online]. Available: `https://arxiv.org/abs/1412.7755`.

[20] K. Xu *et al.*, *Show, attend and tell: Neural image caption generation with visual attention*, 2015. DOI: `10.48550/ARXIV.1502.03044`. [Online]. Available: `https://arxiv.org/abs/1502.03044`.

[21] A. Karpathy and L. Fei-Fei, *Deep visual-semantic alignments for generating image descriptions*, 2014. DOI: `10.48550/ARXIV.1412.2306`. [Online]. Available: `https://arxiv.org/abs/1412.2306`.

[22] A. Dosovitskiy *et al.*, *An image is worth 16x16 words: Transformers for image recognition at scale*, 2020. DOI: `10.48550/ARXIV.2010.11929`. [Online]. Available: `https://arxiv.org/abs/2010.11929`.

[23] W. Liu, S. Chen, L. Guo, X. Zhu, and J. Liu, *Cptr: Full transformer network for image captioning*, 2021. DOI: `10.48550/ARXIV.2101.10804`. [Online]. Available: `https://arxiv.org/abs/2101.10804`.

[24] E. Bruni, G. Boleda, M. Baroni, and N.-K. Tran, "Distributional semantics in technicolor," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 136–145. [Online]. Available: `https://aclanthology.org/P12-1015`.

[25] Y. Feng and M. Lapata, "Visual information in semantic representation," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, California: Association for Computational Linguistics, Jun. 2010, pp. 91–99. [Online]. Available: `https://aclanthology.org/N10-1011`.

[26] R. Socher, M. Ganjoo, H. Sridhar, O. Bastani, C. D. Manning, and A. Y. Ng, *Zero-shot learning through cross-modal transfer*, 2013. DOI: `10.48550/ARXIV.1301.3666`. [Online]. Available: `https://arxiv.org/abs/1301.3666`.

[27] W. Su *et al.*, *Vl-bert: Pre-training of generic visual-linguistic representations*, 2019. DOI: `10.48550/ARXIV.1908.08530`. [Online]. Available: `https://arxiv.org/abs/1908.08530`.

[28] H. Tan and M. Bansal, *Lxmert: Learning cross-modality encoder representations from transformers*, 2019. DOI: `10.48550/ARXIV.1908.07490`. [Online]. Available: `https://arxiv.org/abs/1908.07490`.

[29] W. Kim, B. Son, and I. Kim, *Vilt: Vision-and-language transformer without convolution or region supervision*, 2021. DOI: `10.48550/ARXIV.2102.03334`. [Online]. Available: `https://arxiv.org/abs/2102.03334`.

[30] J. Johnson, M. Douze, and H. Jégou, *Billion-scale similarity search with gpus*, 2017. DOI: `10.48550/ARXIV.1702.08734`. [Online]. Available: `https://arxiv.org/abs/1702.08734`.

[31] M. E. Peters *et al.*, *Deep contextualized word representations*, 2018. DOI: `10.48550/ARXIV.1802.05365`. [Online]. Available: `https://arxiv.org/abs/1802.05365`.

[32] M. Lewis *et al.*, *Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*, 2019. DOI: `10.48550/ARXIV.1910.13461`. [Online]. Available: `https://arxiv.org/abs/1910.13461`.

[33] C. Raffel *et al.*, *Exploring the limits of transfer learning with a unified text-to-text transformer*, 2019. DOI: `10.48550/ARXIV.1910.10683`. [Online]. Available: `https://arxiv.org/abs/1910.10683`.

[34] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, *Electra: Pre-training text encoders as discriminators rather than generators*, 2020. DOI: `10.48550/ARXIV.2003.10555`. [Online]. Available: `https://arxiv.org/abs/2003.10555`.

[35] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[36] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, 2013. DOI: `10.48550/ARXIV.1301.3781`. [Online]. Available: `https://arxiv.org/abs/1301.3781`.

[37] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: `10.3115/v1/D14-1162`. [Online]. Available: `https://aclanthology.org/D14-1162`.

[38] M. Ş. Bilici and M. F. Amasyali, "Variational sentence augmentation for masked language modeling," in *2021 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2021, pp. 1–5. DOI: `10.1109/ASYU52992.2021.9599089`.

[39] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, *End-to-end object detection with transformers*, 2020. DOI: `10.48550/ARXIV.2005.12872`. [Online]. Available: `https://arxiv.org/abs/2005.12872`.

[40] J. Chen *et al.*, *Transunet: Transformers make strong encoders for medical image segmentation*, 2021. DOI: `10.48550/ARXIV.2102.04306`. [Online]. Available: `https://arxiv.org/abs/2102.04306`.

[41] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web.," Stanford InfoLab, Technical Report 1999-66, Nov. 1999, Previous number = SIDL-WP-1999-0120. [Online]. Available: `http://ilpubs.stanford.edu:8090/422/`.

[42] Y. Qiao, C. Xiong, Z. Liu, and Z. Liu, *Understanding the behaviors of bert in ranking*, 2019. DOI: `10.48550/ARXIV.1904.07531`. [Online]. Available: `https://arxiv.org/abs/1904.07531`.

[43] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2014. DOI: `10.48550/ARXIV.1412.6980`. [Online]. Available: `https://arxiv.org/abs/1412.6980`.

[44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826. DOI: `10.1109/CVPR.2016.308`.

[45] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, 2014. DOI: `10.1162/tacl_a_00166`. [Online]. Available: `https://aclanthology.org/Q14-1006`.

[46] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. DOI: `10.48550/ARXIV.1512.03385`. [Online]. Available: `https://arxiv.org/abs/1512.03385`.

[47] I. Loshchilov and F. Hutter, *Decoupled weight decay regularization*, 2017. DOI: `10.48550/ARXIV.1711.05101`. [Online]. Available: `https://arxiv.org/abs/1711.05101`.

[48] A. Agrawal *et al.*, *Vqa: Visual question answering*, 2015. DOI: `10.48550/ARXIV.1505.00468`. [Online]. Available: `https://arxiv.org/abs/1505.00468`.

[49] C. Schuhmann *et al.*, *Laion-400m: Open dataset of clip-filtered 400 million image-text pairs*, 2021. DOI: `10.48550/ARXIV.2111.02114`. [Online]. Available: `https://arxiv.org/abs/2111.02114`.

[50] J. Tiedemann and S. Thottingal, "OPUS-MT — Building open translation services for the World," in *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal, 2020.

Curriculum Vitae

FIRST MEMBER

Name-Surname: Mesut Şafak BILICI
Birthdate and Place of Birth: 24.08.1999, Gaziantep
E-mail: m.safak.bilici@gmail.com
Phone: 0553 385 32 45
Practical Training: Scoutium - Data Science Intern
Artiwise - Natural Language Processing Engineer Intern

SECOND MEMBER

Name-Surname: Enes Sadi UYSAL
Birthdate and Place of Birth: 25.02.1999, İstanbul
E-mail: enessadi@gmail.com
Phone: 0543 938 24 76
Practical Training: Baykar - AI Researcher Intern
DAIMIA - Computer Vision Engineer Intern
YTU CE Probabilistic Robotics Group - Intern

Project System Informations

System and Software: Ubuntu 20.04, Windows 11, Python (and related packages expressed in Section 3)
Required RAM: 16GB
Required Disk: 15GB