REPUBLIC OF TURKEY

YILDIZ TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

AUTOMATED TEXT GENERATION FROM IMAGES BASED ON DEEP LEARNING

20011044 — Yusuf Enes KURT

20011045 — Muhammed Ali LALE

SENIOR PROJECT

Advisor

Prof. Dr. Banu DİRİ

June, 2024

ACKNOWLEDGEMENTS

First of all, we would like to extend our endless thanks and respect to our dear advisor Prof. Dr. Banu Diri, who guided us and comforted us with her support in the selection of the project, the promotion of the project, and the various problems we encountered during the project.

We would also like to thank our professors at Yıldız Technical University Computer Engineering Department who have contributed a great deal to our success.

Yusuf Enes KURT
Muhammed Ali LALE

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

AUTOMATED TEXT GENERATION FROM IMAGES BASED ON DEEP LEARNING

Yusuf Enes KURT
Muhammed Ali LALE

Department of Computer Engineering
Senior Project

Advisor: Prof. Dr. Banu DİRİ

This paper presents a system that takes into account the visual features of images and generates image descriptions based on this information.

Deep learning and natural language processing approaches inspired by the human nervous system are used to train the model. Transformer model is used for image preprocessing, object-word detection and sentence generation.

This project is a work that can benefit visually impaired individuals and social media platforms, and facilitate access to visual information on the internet.

In this article, the detailed content of Transformer, “CLIP” and “GPT-2” models are used to print text on an image are mentioned.

Keywords: Image captioning, Transformer, deep learning, CNN, LSTM, CLIP, GPT-2, feature extraction, encoder, decoder, text generation

ÖZET

GÖRÜNTÜLERDEN DERİN ÖĞRENMEYE DAYALI OTOMATİK METİN ÇIKARMA

Yusuf Enes KURT
Muhammed Ali LALE

Bilgisayar Mühendisliği Bölümü
Bitirme Projesi

Danışman: Prof. Dr. Banu DİRİ

Bu çalışmada resimlerin görsel özelliklerini dikkate alan ve bu bilgiler ışığında resim açıklamaları üreten bir sistem sunulmaktadır.

Modeli eğitirken insan sinir sisteminden esinlenilen derin öğrenme ve doğal dil işleme yaklaşımları tercih edilmiştir. Resmin ön işlemden geçirme, nesne-kelime tespiti ve cümle üretimi için Transformer modeli kullanılmıştır.

Bu proje; görme engelli bireylere ve sosyal medya platformlarına yarar sağlayabilecek, internette görsel bilgiye erişimde ise kolaylığa yol açabilecek bir çalışmadır.

Yazımızda Transformer'ın ayrıntılı içeriğinden, "CLIP" ve "GPT-2" modellerinin resime metin yazdırmak için nasıl kullanıldığından bahsedilmiştir.

Anahtar Kelimeler: Resim başlıklama, Transformer, derin öğrenme, CNN, LSTM, CLIP, GPT-2, özellik çıkarma, encoder, decoder, metin üretimi

1
Introduction

In this section, the technologies underpinning the project, its objectives, scope and requirements are explained and detailed. In this context, we will focus on what technical challenges our project aims to solve.

1.1 Deep Learning

Deep learning is a approach of machine learning that uses ANN. In recent years, it has achieved groundbreaking success, especially in areas such as image and audio processing. Deep learning models can learn complex features automatically from large amounts of data and abstract these features in layers. This method has the capacity to extract important information by detecting objects and structures in the background, especially in tasks such as automatic text extraction from visual content [1].

1.2 Natural Language Processing

NLP is a subfield of AI that helps computers to understand and process human language. It works with data in the form of text and speech, analyzing the structure, meaning and context of language to enable effective communication between machine and human. NLP technologies are used in many different applications such as translation, sentiment analysis, summarization and text extraction. In text extraction projects from images, the processing and interpretation of the extracted text is carried out with NLP methods [2].

1.3 Image Captioning

Image Captioning is a process that summarizes the content of an image by creating descriptive text that describes it. It combines computer vision and NLP techniques. An Image Captioning model looks at a given image and generates a sentence in natural language that describes it. This process is often performed using deep learning models and used in areas such as automated image narration, content description and accessibility tools. [3]

The importance of Image Captioning technology today is highlighted by its applicability and usefulness in many areas. This technology is especially used in areas such as providing automatic content descriptions on social media platforms, improving image search engines, automatically tagging and organizing visual content. It also plays an important role in describing images for visually impaired individuals, thus facilitating their access to digital content. It is also used in sectors such as education, health, security and e-commerce, contributing to data analysis and user experience improvement [4].

1.4 Project Requirements

In the project, a dataset labelled with sentences is needed to train the model. For this reason, image sets named Flickr8k, Flickr30k and MSCOCO are needed.

For modelling the data with various machine learning and deep learning methods, tools and libraries such as PyTorch, NumPy and Pandas are needed.

Platforms such as Anaconda Navigator, Jupyter Notebook and Colab are needed to use such libraries and to code and implement the necessary algorithms on the model.

1.5 Project Scope

The processes expected from our project:

- Analysing and pre-processing visual data,
- Correctly identifying the objects in the pictures and producing a meaningful sentence containing these objects,
- Evaluating the performance of the system with criteria such as BLEU, METEOR, ROUGE etc. and improving the quality of the system according to the results.

1.6 Project Objectives

- The automatic text extraction from images allows non-digital information to be digitized. This simplifies access to visual data.

- Automated text extraction systems are much faster and less costly than manual text typing. This reduces the processing time of large amounts of data and saves labor.

- It minimizes human error, resulting in higher accuracy rates in text extraction. Automated systems offer consistency and accuracy, especially in repetitive processes.

- For users who have difficulty accessing visual data, access to information is facilitated by the vocalization of texts.

- By processing images from social media accounts, it is easier to summarize accounts and categorize accounts. In this way, application algorithms can be used more easily.

2
Preliminary Examination

For the successful realization of our project, it is of great importance to examine similar works and existing methodologies in depth. In this section, important studies in the field of image captioning, the basic approaches used and how these approaches work will be discussed.

2.1 Related Works

There is a study that translates an English image-text dataset into Turkish using automatic translation tools and trains the model on this Turkish dataset to show the Turkish captioning result performance [5].

When the CLIP image encoder and BERT subtitle generator models were used together, they performed well on the MS COCO and Flickr30K datasets [6].

There is a study that discuss the contribution of RNN-based models in image captioning and how to integrate these models into real life [7].

Transformer models emphasize the object-space relationship in images and better understand and interpret the relationship between objects. These models can create more detailed captions [8].

Looking at the work in the MSCOCO image captioning competition, we can better compare human evaluation of image captioning systems with BLEU scores [9].

2.2 Approaches

Work based on image captioning can be divided into four categories: template-based, retrieval-based, generation-based, transformers-based.

2.2.1 Template-Based Image Captioning

This approach generates captions for images using predefined fixed templates or patterns. This method works by identifying objects, actions, and properties in the image to populate a specific template.

2.2.2 Retrieval-Based Image Captioning

This approach finds the most appropriate caption from a large caption data set and matches the selected caption with the image.

2.2.3 Generation-Based Image Captioning

This approach analyzes visual features with deep learning. It generates a caption from scratch according to this analysis.

2.2.4 Transformers-Based Image Captioning

This approach uses transformer architecture and generates captions for images. Very successful results have been obtained with the inclusion of large language models such as ViT, GPT-2 and GPT-3.

2.3 Datasets

The TasvirEt dataset was presented in a paper published by Hacettepe University in 2016 [10]. In this dataset, Turkish descriptions of the photos in the Flickr8k dataset were made by people. For each photo in this dataset, 2 pieces were created. There are about 8 thousand photos and 16 thousand Turkish texts in the dataset. Since this dataset was created entirely by humans, the sentences are meaningful and completely match the image.

The Turkish MSCOCO dataset was presented in 2017 in the paper [5]. MSCOCO contains about 84 thousand photos and there is a dataset with 5 English description sentences for each photo. Turkish MSCOCO translated this ready-made English dataset using translate API. This means approximately 420 thousand translated

Turkish sentences. Although the data is translated, the model gave good results with this data.

3
Feasibility

In this section, the technical, time, legal and economic feasibility of our project will be assessed. In this context, the adequacy of the software and hardware resources to be used, the feasibility of completing the project within the time frame, compliance with legal requirements and economic sustainability will be examined.

3.1 Technical Feasibility

In this section, the technical, time, legal and economic feasibility of the project are discussed.

3.1.1 Software Feasibility

Python was used as the programming language for the project. Python was chosen for its simplicity and readability and it is widely used in machine learning and image processing projects. Virtual environments were created and utilized using Anaconda, designed for data science and machine learning projects. Thanks to these, a powerful and flexible software ecosystem was created. Along with auxiliary libraries such as Numpy, Pandas, PyTorch, Tensorflow libraries were used specifically for natural language processing techniques. The fact that all of these libraries are open source and freely available was also taken into consideration during library selection.

3.1.2 Hardware Feasibility

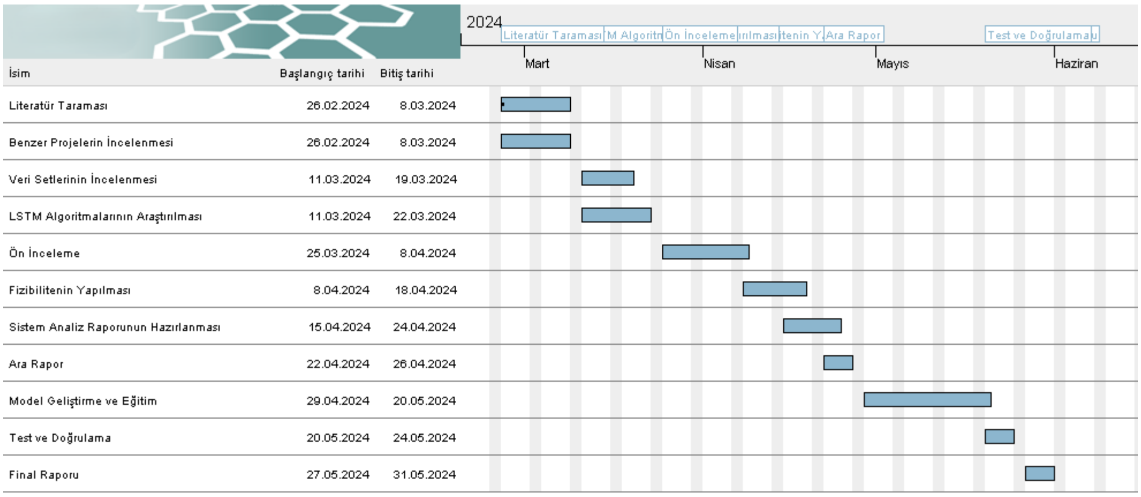
Monster Abra A7 and Lenovo Ideapad Gaming 3 Model computers were used as hardware components for our project. Both computers have 16 GB RAM and 1 TB SSD capacity. The GPUs of the computers are NVIDIA GeForce GTX1650 with 4 GB VRAM and have Ryzen 5 5600H and Intel I5 10300H CPUs. The computers have Windows 11 Pro (6400TL) operating system, but the Pro version is not required and was not included in the feasibility assuming that other computers have Windows.

Since the project requires the presence of systems with high GPUs and TPUs, the Google Colab environment was chosen to compile and run the project, which allows faster processing with graphics cards with different processing powers such as T4, V100, A100, etc., since it would be inefficient to use the computers' own processors and graphics units during the train phase. The monthly cost is 165,60TL. Google Colab or the computers' own processors can also be used in the evaluate phase to evaluate the results.

In the presence of computers with higher GPU and CPU power, there will be no need to use Google Colab environment. However, computers with lower processing power can also be used when using Colab.

3.2 Time Feasibility

The time required to complete the planned works is shown in Figure 3.1.



İsim	Başlangıç tarihi	Bitiş tarihi
Literatür Taraması	26.02.2024	8.03.2024
Benzer Projelerin İncelenmesi	26.02.2024	8.03.2024
Veri Setlerinin İncelenmesi	11.03.2024	19.03.2024
LSTM Algoritmalarının Araştırılması	11.03.2024	22.03.2024
Ön İnceleme	25.03.2024	8.04.2024
Fizibilitenin Yapılması	8.04.2024	18.04.2024
Sistem Analiz Raporunun Hazırlanması	15.04.2024	24.04.2024
Ara Rapor	22.04.2024	26.04.2024
Model Geliştirme ve Eğitim	29.04.2024	20.05.2024
Test ve Doğrulama	20.05.2024	24.05.2024
Final Raporu	27.05.2024	31.05.2024

Figure 3.1 Gantt Diagram

3.3 Legal Feasibility

The dataset we use is a free and accessible dataset. There are no legal obstacles to the use of these datasets in our project. The libraries used with the Python language in the project are also open source and free. There is no factor that would violate any law or regulation in the project.

3.4 Economic Feasibility

The monthly fee of Google Colab, the development environment used, is 165,60TL. The libraries and frameworks we use are free. The dataset we use is also free.

The cost table is shown in Table 3.1.

Table 3.1 Economic Feasibility

Description	Qty/time	Price	Total
Google Colab Pro	3 months	165,60TL	496,80TL
Language and libraries used	3 months	0TL	0TL
Lenovo Ideapad Gaming 3	1	25.000TL	25.000TL
Monster Abra A7	1	25.000TL	25.000TL

4
System Analysis

In this section, the components of the system, requirements, objectives and performance metrics of our project will be discussed in detail.

4.1 Requirements

- Visual features should be extracted from the images.
- Objects in images should be detected and captions should be generated with these objects.
- Image processing and natural language processing techniques should be used together using deep learning algorithms algorithms. Using these techniques, an image captioning model should be designed and trained using features.
- An image should be tested on the trained system and take an output with captions.

4.2 Objectives

Our primary objectives in our work are the following:

- Tasks such as object recognition should be performed by examining the images and the images should be analyzed correctly.
- Meaningful sentences should be produced for the images.
- These sentences should be related to the objects in the images and explain this scene.

4.3 Performance Metrics

Image captioning models use some common metrics to measure the accuracy of the captions produced. These metrics include accuracy, recall, precision. Some common metrics:

4.3.1 BLEU

A widely used statistic, BLEU evaluates how accurate the generated captions are compared to the ground truth of the dataset. The agreement between predicted and expected captions is indicated by a score, denoted as BLEU score. A higher score means a higher match [11].

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{i=1}^{N} w_i \log(p_i)\right) \tag{4.1}$$

4.3.2 METEOR

Another metric, METEOR, is designed to address some of the problems found in the more popular BLEU metric. It has features not found in other metrics, such as exact word matching as well as root and synonym matching [12].

$$METEOR = F_{\text{mean}}(1 - p) \tag{4.2}$$

4.3.3 ROUGE-L

ROUGE is a metric used in natural language processing to compare machine-generated text with real text as a reference. ROUGE metrics range from 0 to 1, with a higher score indicating a higher agreement.

$$\text{ROUGE-L} = \left(\frac{\text{LCS}}{\text{Reference\_length}}\right) \times 100 \tag{4.3}$$

ROUGE-L also calculates the length of common sub-indexes with the LCS (Longest Common Subsequence) problem and calculates how many n-grams the reference and prediction results have in common [13].

5
System Design

Our project focuses on detecting objects and actions in images. For these identified objects and actions, captions those containing the image summary are generated using NLP methods. The CLIP model was used to convert image features into vectors and to associate objects in the image with captions. The GPT-2 model is used to generate meaningful and appropriate captions for the image. In addition, the CLIP model uses CNN for visual feature analysis.

5.1 Neural Networks

Neural networks are a study of AI inspired by the nerves in the human brain. They are used to process and learn complex data. Input, hidden and output layers are the three main layers of these neural networks.

5.1.1 CNN

A neural network structure called CNN is designed for image processing and image recognition. It is especially used for working with 2D or 3D data. They contain convolution layers specialized for recognizing the patterns of the image. These layers perform the necessary filtering operations.

CNN is used in our project to extract features from images. These models can detect various patterns, edges and texture information in the image. Details such as objects, colours and shapes in an image are examples of these. These details are analysed by CNN and these analyses are transferred to the next language processing stage.

CNN models have multiple layers. Each layer is designed to learn a specific feature set from the image. Thanks to these layers, it is ensured that the image is understood in depth and more accurate sentences are produced.

Once trained, CNN models can process new images quickly and effectively. This helps "Image Captioning" systems to achieve high performance even in real-time applications.

5.2 Transformer

The Transformer concept entered the literature with Google's article "Attention Is All You Need" published in 2017. It is a revolutionary artificial intelligence model in natural language processing. The main features of the Transformer approach model are:

Attention Mechanism: When processing information in features, Transformer focuses on the entire data simultaneously. In pre-Transformer models, this was done sequentially. But with this approach, Transformer makes it easier to focus on important parts of the data. Thus, it can better understand the relationship between different parts of the data.

Parallel Process: Transformer shortened the model training time by processing the data in parallel rather than sequentially. This enables the use of larger datasets.

Flexibility: Transformer covers a wide range of NLP tasks such as word sequence from image, word sequence from word sequence, vector models from word sequence [14].

The transformer architecture is shown in Figure 5.1.

Thanks to its attention mechanism, Transformer helps the model to focus on important areas in an image and determine how these areas are associated with language output. For example, if there is a dog and a ball in a photograph, the Transformer model uses this mechanism to understand the relationship between the words 'dog' and 'ball' and how they are linked to their visual representations.

5.3 CLIP as Image Encoder

5.3.1 Reason for Choosing CLIP

CLIP can process text and images simultaneously. Thanks to this feature, it produces more accurate results by combining images and their descriptions. The reason for using this model in our project is to better generate text that describes the images.

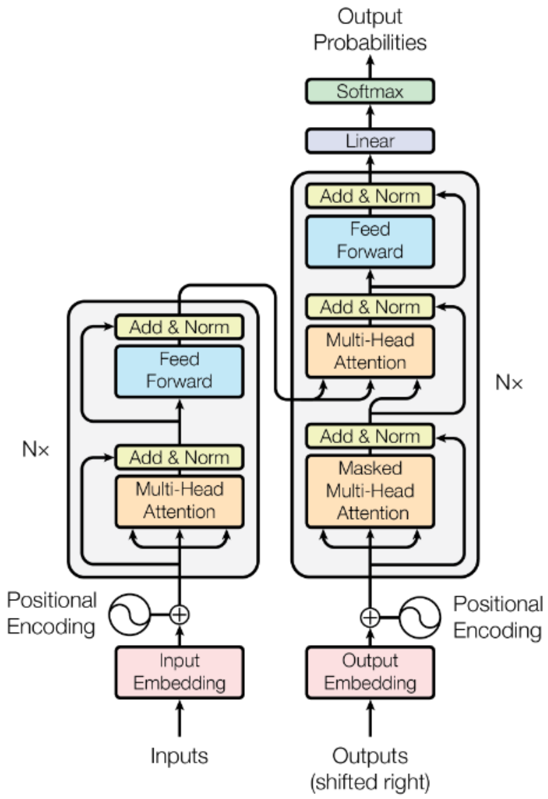


Figure 5.1 Transformer Architecture [14]

5.3.2 Working Principle of CLIP

CLIP is a pretraned model on a large dataset. It evaluates images and captions of these images together. In this way, the model can match the content of an image with the appropriate caption. In our project, CLIP uses the features it receives from the image to generate appropriate captions.

5.3.3 Implementation of CLIP in the Project

In our project, CLIP analyzes the incoming visual data and extracts feature vectors. The information obtained from the images is fed into the GPT-2 model to create meaningful and appropriate captions. This method provides high accuracy and consistency in image recognition and text generation.

5.4 GPT-2 for Text Generation

5.4.1 Reason for Choosing GPT-2

GPT-2 is a very successful artificial intelligence model for text generation. This model has a wide language modeling capacity and can generate coherent text on a variety of topics. The main reason for using GPT-2 in our project is to describe images with descriptive and accurate text.

5.4.2 Working Principle of GPT-2

GPT-2 is a pretrained model on a large amount of text. This model generates new texts using the language rules and contexts it has learned. In our project, visual feature data from CLIP is fed to this model and GPT-2 generates appropriate texts

5.4.3 Implementation of GPT-2 in the Project

In the project, GPT-2 generates descriptive texts (captions) for these images based on visual features from CLIP. Thanks to the language capability of the model, the generated descriptions express the visual content in an accurate and understandable way.

6
Implementation

In this section, the technical implementation process and operational logic of our project will be detailed. Additionally, our estimation process on a single image, the algorithms used and how the outputs are processed will be analyzed.

6.1 Operating Logic of the Programme

In figure 6.1 you see a visual sketch of the image caption transformer model. Let's explain the diagram step by step:

Input Image: The model gets an input image, the size of this image is defined by a specific batch size, image size and image channels.

CLIP Image Encoder: The input image first passes through the CLIP image encoder. This encoder converts the image into an encoded vector array (encoded image-1). The CLIP encoder consists of a position embedding and repeated blocks of array length. Each block contains a position normalisation (Norm), a multi-head attention mechanism and another normalisation layer, followed by an MLP (multilayer perceptron) and another normalisation layer.

Vision Transformer Block: The encoded image-1 then enters the Vision Transformer (ViT) block. ViT transforms the encoded image into encoded vector array (encoded image-2). This block consists of N number of recurrent layers including multiple head attention mechanism and norm layers, followed by a linear layer, MLP and a norm layer.

Text Decoder: Finally, the encoded image-2 enters the text decoder. This section converts the encoded image vector into an output caption. The text decoder includes attention mechanisms (both cross and self attention), normalisation layers and linear layers. There are also position and text embedding layers, which serve as input to the decoder.

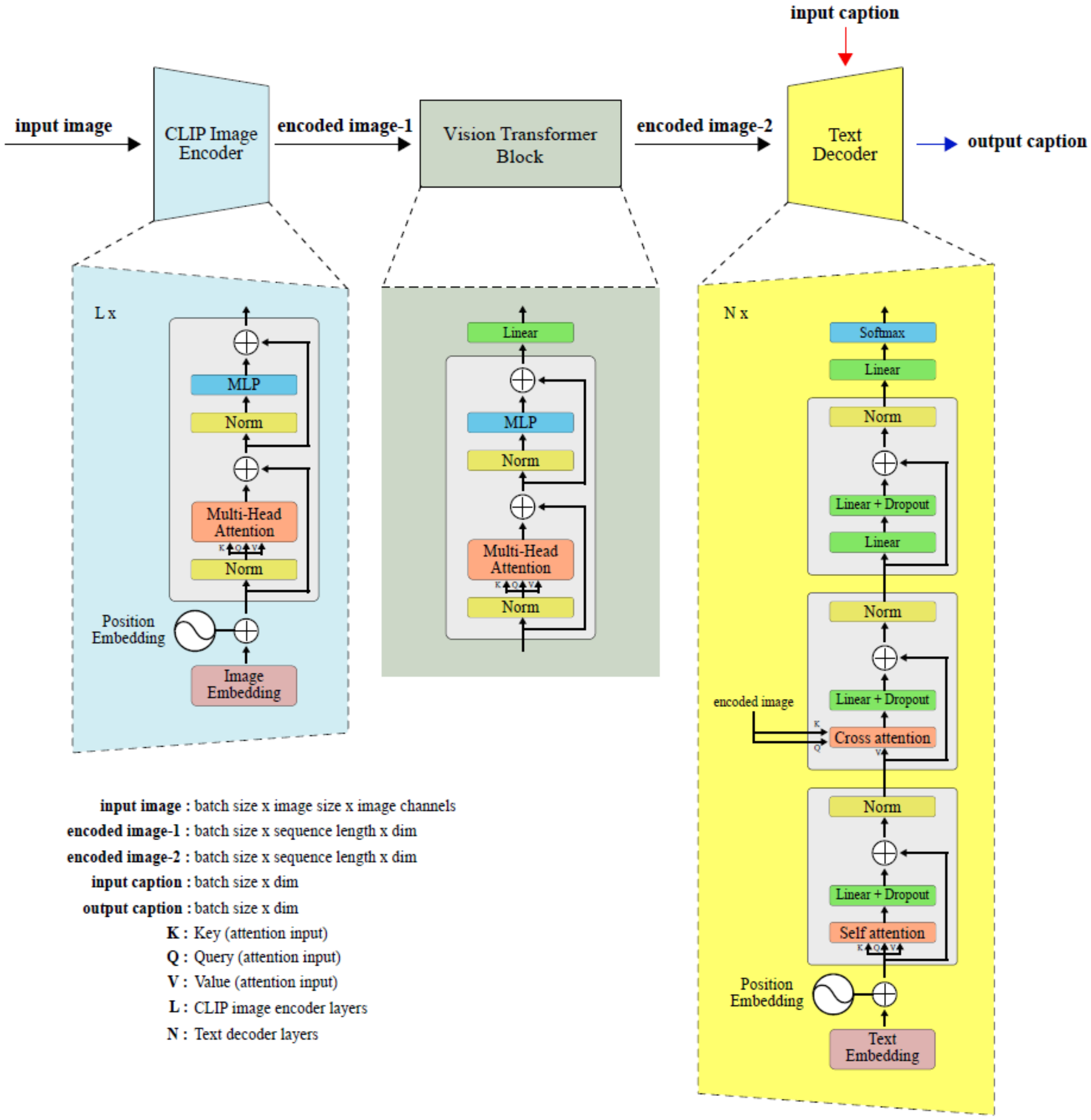


Figure 6.1 Visual scheme of the proposed deep Turkish image captioning model [6]

Output Caption: As a result, the model generates a caption for the input image. This caption text is represented as a vector of size (batch size x dimension).

6.1.1 CLIP Image Encoder

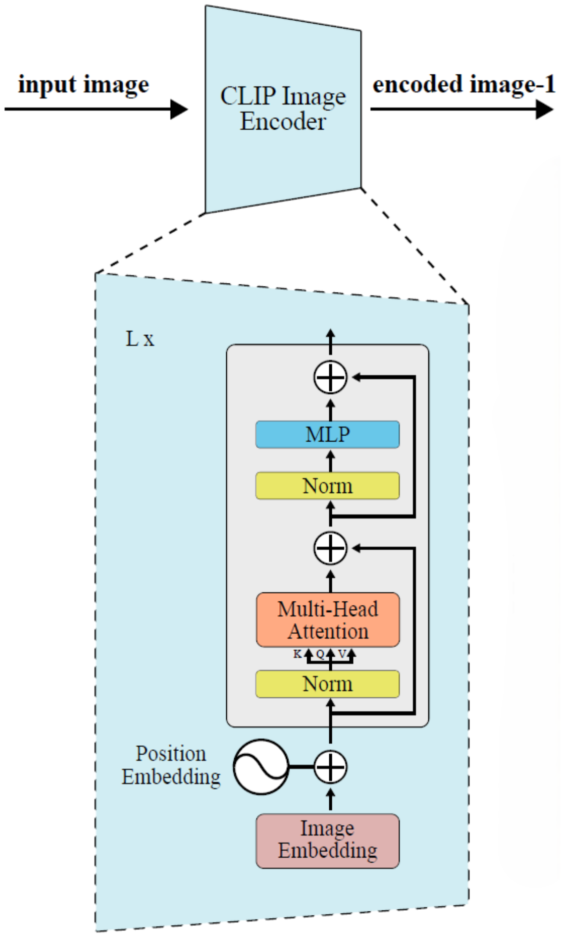


Figure 6.2 CLIP Image Encoder [6]

Figure 6.2 shows the internal structure of the CLIP Image Encoder in detail. Let's explain the process steps one by one:

Image Embedding: The first step is to convert the input image into embedding vectors. The image is converted into a vector of a given size (usually flattened pixels or features obtained from a pre-trained network).

Position Embedding: Position information is added to each embedding vector to enable the model to understand the order of the embedding vectors. This helps the model to understand that the embedding vectors in the array are ordered and to understand their spatial relationship to each other.

Norm: The first norm layer performs the normalisation of the vectors (i.e. scaling with standard deviation 1 and mean 0). This makes the training process more stable and faster.

Multi-Head Attention Mechanism: Here, the embedding vectors perform a series of attention calculations under different "heads". This allows the model to "pay attention" to different parts of the image by assigning different weights to different embedding vectors. At this layer, the attention mechanism has three main elements: query, key and value. The product of query and key creates a weight, which is used to sum the values and produce an attention score.

MLP (Multi-Layer Perceptron): After normalisation, the resulting vectors are passed through one or more dense layers. These layers usually contain activation functions and help the model to learn complex relationships.

Aggregation: The vectors from the MLP are again summed with the output of the previous layer (residual link).

Encoded Image-1: After all these operations, a array of vectors representing the features of the input image of the model is obtained. These are the basic embedding vectors that the model will work with before proceeding to the next stages.

6.1.2 Vision Transformer Block

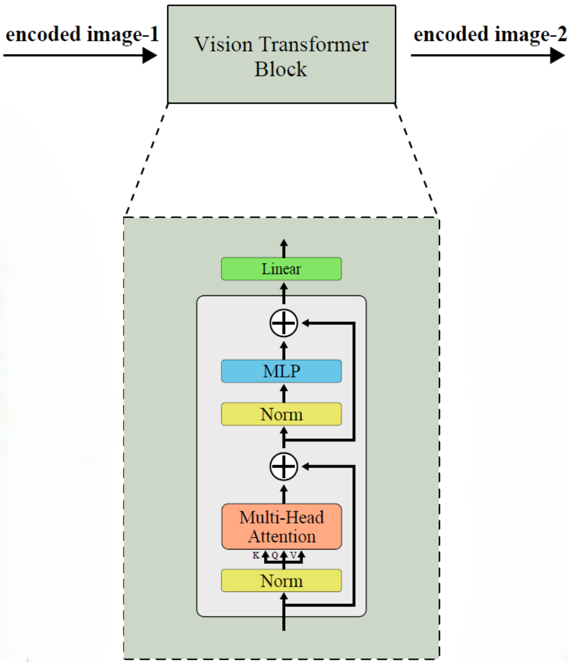


Figure 6.3 Vision Transformer Block [6]

The ViT Block is shown in Figure 6.3. This block takes the encoded image vectors through a deeper processing. Here is a step-by-step explanation:

Encoded Image-1: This block receives the encoded image vectors output from the CLIP Image Encoder.

Norm: The vectors are passed through a normalization layer before they are processed. This allows the data to be scaled and processed more efficiently in the next steps.

Multi-Head Attention: The encoded image vectors are subjected to a multi-head attention mechanism that allows the model to assign different weights to different embedding vectors. This stage decides which parts of the vectors to focus on. Here, the query, key and value vectors are used.

Aggregation and Norm: The output of the attention mechanism is aggregated with the previous vector from the previous normalization and then passed through the norm layer.

Norm: The vectors output from the attention mechanism are re-normalized.

MLP: After normalization, the resulting vectors are passed through a multilayer perceptron (MLP). The MLP consists of one or more dense layers and activation functions and helps to learn more complex features of the vectors.

Aggregation and Norm: The MLP output is summed with the vector from the previous normalization layer and passed through the next normalization process.

Linear Layer: After each transformation, the vectors pass through a linear layer that ensures that the model has an appropriate dimensionality for the next steps.

Coded Image-2: The final coded image vectors coming out of the ViT block are transformed into the final coded image-2 before entering the text decoder.

6.1.3 Text Decoder

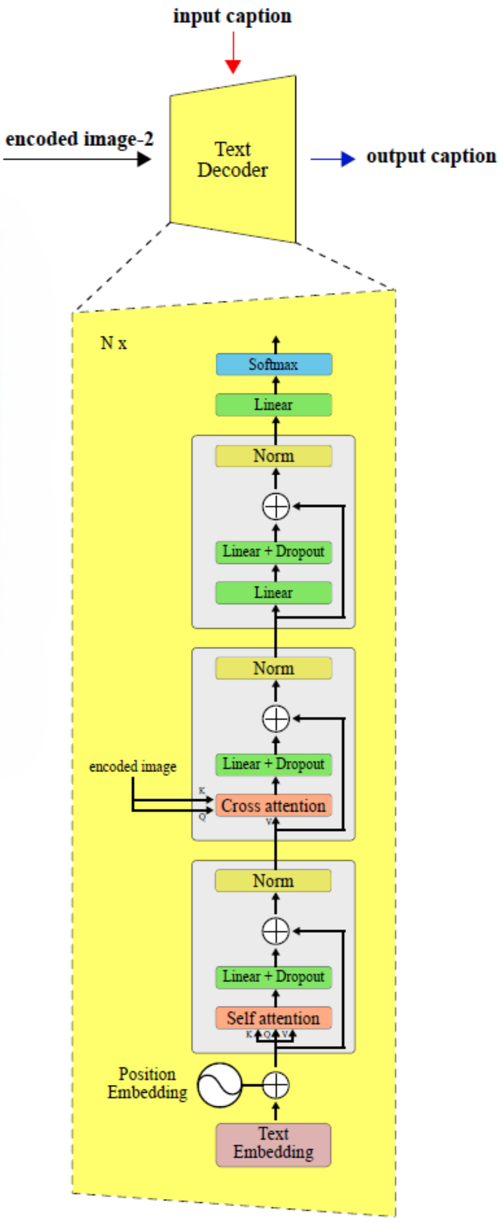


Figure 6.4 Text Decoder [6]

The Text Decoder in Figure 6.4 illustrates the process of generating an output caption using the data from the encoded image-2. Let's analyze the process steps one by one:

Text Embedding: First, a text embedding vector is taken, which serves as the start of the caption that will form the output of the model.

Position Embedding: Position information is added to the text embedding vectors to help the model understand the word order in the caption.

Self Attention Mechanism:

- **Self Attention:** The model uses the self-attention mechanism to understand the context between the words in the caption.
- **Linear + Dropout:** The normalized vectors go through a linear transformation with dropout added. Dropout is used to prevent overfitting.
- **Norm:** Text embeddings are normalized. This standardizes the data, allowing the model to learn more effectively.

Cross Attention:

- **Cross Attention:** A cross attention mechanism is applied between the encoded image vectors and the text embeddings. This allows the model to recognize which part of the image is associated with the caption.
- **Linear + Dropout:** Again a linear transformation and dropout are applied before merging with the encoded image information.
- **Norm:** At this stage, the normalization process is applied before the encoded image is merged with the information from image-2.

Output Layers:

- **Norm and Linear Layers:** After cross-attention, the vectors go through normalization again and are prepared for the next steps by passing through a linear layer.
- **Softmax:** This function is used to calculate the possible next word probabilities for each word of the model.

Output Caption: As a result of all these operations, the model produces a caption associated with the image. This caption is based on the model's understanding of the content of the input image and generates a text describing it.

The Text Decoder thus generates a meaningful caption that contains semantic data about the image, using the grammatical structures learned by the model. This is the process of producing a human-understandable text that describes the image.

6.2 Single Image Prediction

Single image prediction is the process of generating captions for an image. From an image file, a caption is generated with the trained model. This process consists of the following steps.

- Firstly, the visual elements of the image are analysed.
- Then the visual features of the image are extracted with the CNN technique and saved in a file.
- The saved feature file is given to the model and a sentence about the content of the image is created.
- Finally, the meaning of this sentence is checked and its compatibility with the content of the image is compared.

7 Experimental Results

In this section the dataset used in our project, the techniques used and the impact of different evaluation criteria on the success of the project are described.

The results of different experiments with different metrics were analyzed by characterizing the tests performed with the trained models using various metrics.

The biggest factors we changed to observe the impact on the results were the dataset size and the model's decoder (for text generation).

7.1 Dataset

The dataset we used is the MSCOCO dataset containing about 82,783 images. In order to observe the effect of different sizes of data on the results, datasets of 4 different sizes that our resources can handle were created.

For train, 1,000, 8,000, 32,000 and 66,000 images were used. For the test, the rest of the dataset, 16,000 images, was used. In other words, 80% of the dataset was used for train and 20% for test. For text generation, 2 different GPT-2 models were used.

7.2 Language Models

In our project, we used two different GPT-2 models to text generation in Turkish language: "ytu-ce-cosmos/turkish-gpt2" and "redrussianarmy/gpt2-turkish-cased". Both models are designed for Turkish text generation and language modeling. “ytu-ce-cosmos/turkish-gpt2” model was trained on a larger dataset and may have a better chance of success. On the other hand, the “redrussianarmy/gpt2-turkish-cased” model can work with different programming tools such as PyTorch and TensorFlow. this allows for more flexible working environments.

7.2.1 ytu-ce-cosmos/turkish-gpt2

This model was developed by the Cosmos research group at Yıldız Technical University. It was trained using a 250 GB CulturaX dataset and an additional 25 GB of data. This data was collected from web resources, books, forums and news sites.

7.2.2 redrussianarmy/gpt2-turkish-cased

This model, trained using the Oscar-corpus dataset, works on Turkish texts. The model is trained using byte-level BPE tokenization. It is compatible with both PyTorch and TensorFlow.

7.3 Results

The comparison of the accuracy of the trained models with different dataset size and different text generation models with various metrics is given in Table 7.1.

Table 7.1 Comparative Performance Metrics of Machine Translation Models on Different Dataset Sizes

Model	Dataset	BLEU_3	BLEU_4	METEOR	ROUGE-L	CIDEr
redrussianarmy	1k	4.7385	3.7673	14.0990	23.1137	
redrussianarmy	8k	5.4766	4.3209	15.7892	25.3604	
redrussianarmy	32k	6.2741	4.9229	17.0218	26.9002	
redrussianarmy	66k	6.3011	4.9521	17.3063	27.2205	
ytu-ce-cosmos	1k	3.5741	2.8942	11.3198	18.3660	
ytu-ce-cosmos	8k	5.7983	4.5473	16.3455	26.1227	
ytu-ce-cosmos	32k	6.2213	4.8890	17.1934	26.9228	
ytu-ce-cosmos	66k	6.3738	5.0113	17.5137	27.2593	

8
Performance Analysis

In this section, the results in Table 7 are analyzed according to particular variables.

8.1 Dataset

We created 4 different size datasets from the MSCOCO dataset: 1k, 8k, 32k and 66k. The results of the models trained using these datasets measured with BLEU, ROUGE and METEOR metrics are shown in Table 7.1. According to these results, it is observed that as the dataset size increases, the performance of the trained models also increases. The same situation was observed for all metrics used.

However, while there are large increases until the 32k data set, there is no significant performance difference between 32k and 66k data sets.

8.2 Language Models

In the project, datasets of different sizes were trained with 2 different GPT-2 models.

When the results in Table 7 are analyzed, we observe a significant difference only in the 1k models. Among the models trained with the 1k dataset, the model trained with "redrussianarmy/gpt2-turkish-cased" scores much higher than "ytu-ce-cosmos/turkish-gpt2" in all metrics.

In the 8k and 66k datasets, the "ytu-ce-cosmos/turkish-gpt2" model is slightly better, while in the 32k dataset, the "redrussianarmy/gpt2-turkish-cased" model performance is slightly higher.

Considering these results, we see that none of the GPT-2 models has a clear advantage over the other.

8.3 Results

The results show that the dataset size has a direct proportional effect on performance. But after a certain point this difference has decreased. However, the GPT-2 models used did not make a significant difference.

9
Results

Curriculum Vitae

FIRST MEMBER

Name-Surname: Yusuf Enes KURT
Birthdate and Place of Birth: 24.11.2000, İstanbul
E-mail: enes.kurt1@std.yildiz.edu.tr
Phone: 0551 112 98 59
Practical Training:

SECOND MEMBER

Name-Surname: Muhammed Ali LALE
Birthdate and Place of Birth: 30.11.2002, Yozgat
E-mail: ali.lale@std.yildiz.edu.tr
Phone: 0534 549 29 66
Practical Training:

Project System Informations

System and Software: Windows Operating System, Python
Required RAM: 12GB
Required Disk: 20GB