2022

M.Sc. in Electronics and Computer Engineering

Juman SAKKAR

HASAN KALYONCU UNIVERSITY

GRADUATE SCHOOL OF

NATURAL AND APPLIED SCIENCES

Efficient Image Annotation and Caption System Using Deep Convolutional Neural Networks

M. Sc. THESIS

IN

ELECTRONICS AND COMPUTER ENGINEERING

BY

Juman SAKKAR

JANUARY 2022

HASAN KALYONCU UNIVERSITY

GRADUATE SCHOOL OF

NATURAL AND APPLIED SCIENCES

Efficient Image Annotation and Caption System Using Deep Convolutional Neural Networks

M.Sc. THESIS

IN

ELECTRONICS AND COMPUTER ENGINEERING

BY

Juman SAKKAR

2022

Efficient Image Annotation and Caption System Using Deep Convolutional Neural Networks

M.Sc. Thesis

in

Electronics and Computer Engineering

Hasan Kalyoncu University

Supervisor

Asst. Prof. Dr. Saed ALQARALEH

By

Juman SAKKAR

2022

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES INSTITUTE M.Sc. ACCEPTANCE AND APPROVAL FORM

Electronics and Computer Engineering Department, Electrical and Electronics Engineering M.Sc. (Master of Science) programme student **Juman SAKKAR** prepared and submitted the thesis titled **Efficient Image Annotation and Caption System Using Deep Convolutional Neural Networks** defended successfully on the date of **14/01/2022** and accepted by the jury as a M.Sc. thesis

Position	**Title, Name and Surname Department/University**	**Signature**
Supervisor	Asst. Prof. Dr. Saed ALQARALEH Computer Engineering Department Hasan Kalyoncu University	
Jury Member	Prof. Dr.Muhammet Fatih HASOĞLU Computer Engineering Department Hasan Kalyoncu University	
Jury Member	Dr. Öğr. Üye. Bülent HAZNEDAR Computer Engineering Department Gaziantep University	

This thesis is accepted by the jury members selected by the institute management board and approved by the institute management board.

Prof. Dr. ……

Director

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Juman SAKKAR

ABSTRACT

Efficient Image Annotation and Caption System
Using Deep Convolutional Neural Networks
Juman SAKKAR

M.Sc. in Electronics and Computer Engineering.

Supervisor: Dr. Saed ALQARALEH

2022, 60 pages

In recent years, with the advances in the artificial intelligence field, image annotation also known as image description (IAC) has progressively attracted researchers' attention. IAC automatically creates natural text descriptions according to the image contents. IAC combines the knowledge of computer vision and natural language processing.

In this research, a novel image annotation and description system was developed. The main parts of the developed system are Convolution Neural Network (CNN) and Long Short Time Memory (LSTM). Also, the developed system was enhanced by multiple steps such as adding regularizing to convolution layers, adding dropout layers to the fully connected layers, using genetic algorithms to find the most suitable batch size, and investigating the performance of multiple optimizers such as Adaptive Moment Estimation (Adam), Stochastic Gradient Descent(SGD), and Nesterov accelerated gradient to find the most suitable one for the developed approach.

The developed system was validated by multiple experiments using one of the challenging datasets, i.e., the Flicker dataset. Overall, our improved model outperformed the existing state of arts using the BLEU metric. Also, results prove that the designed system can effectively describe images. Last but not least, this research help researchers by highlighting some open challenges in the field of image annotation.

Key words: Aartificial Intelligence, Image Annotation, Convolution Neural Network, Long Short Time Memory.

ÖZET

Verimli Görüntü Açıklama ve Altyazı Sistemi
Derin Evrişimli Sinir Ağlarının kullanımı
Juman SAKKAR

Yüksek Lisans Tezi, Elektronik Bilgisayar Müh. Bölümü

Tez Yöneticisi: Dr. Saed ALQARALEH

2022, 60 Sayfa

son yıllarda, yapay zeka alanındaki gelişmelerle birlikte, görüntü açıklaması (IAC) olarak da bilinen görüntü açıklama, araştırmacıların ilgisini giderek daha fazla çekmiştir. IAC, görüntü içeriğine göre otomatik olarak doğal metin açıklamaları oluşturur. IAC, bilgisayarla görme ve doğal dil işleme bilgilerini birleştirir.

Bu araştırmada, yeni bir görüntü açıklama ve açıklama sistemi geliştirilmiştir. Geliştirilen sistemin ana parçaları Evrişim Sinir Ağı (CNN) ve Uzun Kısa Süreli Bellek (LSTM)'dir. Ayrıca geliştirilen sistem, evrişim katmanlarına düzleştirme ekleme, tamamen bağlı katmanlara bırakma katmanları ekleme, en uygun parti boyutunu bulmak için genetik algoritmaları kullanma ve uyarlamalı moment gibi çoklu optimize edicilerin performansını inceleme gibi birçok adımla geliştirilmiştir. Tahmin (Adam), Stochastic Gradient Descent (SGD) ve Nesterov hızlandırılmış gradyan, geliştirilen en uygun olan yaklaşımı bulmak için.

Geliştirilen sistem, zorlu veri kümelerinden biri, yani Flicker veri kümesi kullanılarak birden fazla deneyle doğrulandı. Genel olarak, geliştirilmiş modelimiz BLEU metriğini kullanarak mevcut son teknolojiden daha iyi performans gösterdi. Ayrıca sonuçlar, tasarlanan sistemin görüntüleri etkili bir şekilde tanımlayabildiğini kanıtlamaktadır.

Son olarak, bu araştırma, görüntü açıklamaları alanındaki bazı açık zorlukları vurgulayarak araştırmacılara yardımcı olur.

Anahtar Kelimeler: Yapay Zeka, Görüntü Açıklaması, Evrişim Sinir Ağı, Uzun Kısa Zamanlı Bellek.

To My Family

ACKNOWLEDGEMENTS

First and foremost, I want to thank Allah, the Almighty, the Most Gracious, and the Most Merciful, for the blessings He has bestowed upon me throughout my studies and in the completion of this thesis.

I would like to express my gratitude and sincere thanks to my supervisor Saed ALQARALEH, for his invaluable knowledge, experience and assistance that were crucial in carrying out my research and completing this thesis.

My sincere gratitude also goes to the honorable members of the Discussion Committee, Prof. Dr. Muhammet Fatih Hasoglu and Dr. Öğr. Üye. Bülent HAZNEDAR, for their valuable advice and comments, besides granting me their approval to discuss this thesis.

In addition, I would like to thank my great parents, supportive husband and my loving siblings for their endless support, love, and prayers.

Finally, I am sincerely grateful to all those who supported me throughout my work and contributed to the completion of this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

IAC	Image Annotation and Caption
ADA	Americans with Disabilities Act
NAD	National Association of the Deaf
IA	Image Annotation
CNN	Convolution Neural Network
ANN	Artificial Neural Network
LSTM	Long Short Term Memory
MEF	Manually Engineering Features
ICNN	Integrated Convolutional Neural Network
DNN	Deep Neural Networks
ML	Machine Learning
NLP	Natural Language Processing
GRU	Gated Recurrent Units
NN	Neural Network
GA	Genetic algorithm
EA	Evolutionary Algorithms
Adam	Adaptive Moment Estimation
SGD	Stochastic Gradient Descent
BLEU	Bilingual Evaluation Understudy Score

CHAPTER 1

INTRODUCTION

In this chapter, an introduction to the image annotation systems is presented. First, we begin with a background that provides the basic vocabulary for the research problem. Then, the research objectives are defined. A plan with the required steps to achieve the research objectives is presented and discussed. Finally, an overview of research on image annotation is provided highlighting the distinction between our work and other research.

1.1 Background

Automatic generation of image captions or descriptions is a challenging task that needs both visual information and linguistic knowledge. In other words, it requires not only complete image understanding but also natural language processing and sophisticated natural language generation. This is why it is considered such an interesting challenge that has been adopted by both the computer vision and natural language processing communities [1].

Initially, it was assumed that it was impossible for a bot such as a computer to describe or summarize an image. With the advancement of techniques that can be used for such tasks, such as deep learning and the availability of a huge amount of data (annotated images), we could build some models that can generate captions and/or descriptions of images.

The main challenge related to image annotation can be image processing. This challenge is also a computer vision task. It is worth mentioning that this problem also deals with other steps, such as creating textual descriptions. To sum up, there are mainly two phases to handling image annotation:

- Feature extraction stage, each image should be summarized by a vector of numerical values.
- Annotation generating stage, that handles the input (the input is the feature vector), which works on producing a combination of words in logical ordering, i.e., significant caption or description for the input image.

1.2 Problem Statements

Nowadays, it is obvious that the task of Image Annotation and Caption (IAC) is outlined using deep learning techniques to produce an efficient and accurate system that can handle both types of data: images and text. In this study, we aim to improve existing IAC solutions not only by adapting the Deep Convolutional Neural Networks (ConvNet) but also by investigating the effect of combining multiple machine learning approaches.

We want to build a system that can automatically describe the content of an image in Natural Language (plain English). The system should be able to generate novel descriptions that are both diverse and of high quality.

Nowadays, the number of researchers interested in automatically generating image captions to describe image context is increasing, as automated captioning will save both cost and time when captioning is done by human users.

An image captioning system is a challenging task. Since images might contain multiple types of data and objects, converting this data into understandable sentences is a useful task, but hard at the same time. The resulting captions should be logical and real, with no fake information. The challenging task is how to attach each object to the right activity in the image. The main challenges are summarized below:

- An interdisciplinary problem that combines computer vision and NLP to form new problem fields. In other words, image captioning requires using and adapting efficient techniques from Computer Vision and NLP.
- Expressing semantic relations between image's objects and the activities they are involved in. This is not only an issue of annotations, it is also related to describing the images.
- It is extremely hard to evaluate how well the annotation system performs, as even in real life it is hard and various descriptions can be assigned to the same image.

1.3 The Possibility Of Realizing Automatic Caption

Automatically captioning image content, aided by natural language, is an essential and challenging task. With advancements in calculation techniques and the existence of very large datasets, it is now possible to build and design models that have the ability to

generate captions for an image. On the other hand, humans are able to easily describe the environments they are in and/or any images they see. Hence, it is normal for any human to talk and describe the details of any image quickly. In addition, with the great evolution in data science and computer vision fields, some challenges such as attribute classification, scene recognition, image classification, recognizing an object, and action classification are possible.

1.3.1 **Applications of Automatic Image Caption**

Captioned images have been a rising trend in many fields and are currently used for multiple purposes. For instance, the Americans with Disabilities Act (ADA) and the National Association of the Deaf (NAD) are adapting automatically captioned images to achieve their goals. In the following, we will summarize some other fields that use captioned images.

- Communications: automatic captions convey vital information about who is doing what, when, where and (sometimes) why. Solid captions paired with interesting photographs can spark a reader's interest in a full-text story. Without captions, people draw their own conclusions about a photo, so automatic captions unify opinions between people and companies without drawing the biased conclusions of individuals.
- Online learning: inserting images that summarize subjects in online learning sounds like a good idea to save teachers' time and deliver knowledge faster. Another subfield is teaching children: children of various ages may find that selecting random images and creating captions for these images helps them understand more and learn new things.
- Video annotation: it is an advanced idea where the automated generated caption not only describes objects and activities in an image or in a video, but also gets the target of the image or video.
- On social media, when users are uploading images without details, a captioning tool may help in describing such files and even showing such images to related friends or viewers.

1.4 Research Objectives

This thesis aims to develop a multi-objective system that accurately annotates images. Moreover, we aim to improve the performance of existing image annotation techniques. To achieve our goal, the Deep Convolutional Neural Networks (ConvNet) are integrated to build the desired efficient system. Of course, thanks to the high classification performance of deep features, we believe this can significantly improve the overall performance of proposed systems. To achieve these objectives, the following steps are considered and performed:

1. Investigate different Convolution Net models to find the best one. After that, integrate and train the selected one. It is worth mentioning that the selected Convolution Net architecture will be used for image annotation.
2. Investigate the effect of using state-of-the-art word embedding and feature extraction algorithms.
3. Use the Genetic Algorithm to tune the used model parameters and improve the performance of CNN in general and the developed approach in particular.
4. Validate our system approach on challenging datasets such as Flickr8k and compare its performance with some state-of-the-art systems.

1.5 Related Works

In this section, the most recent and related state-of-the-art studies are summarized. In [2], a new picture subtitling model dependent on undeniable level picture highlights was proposed. Here, Both low-level data, and significant level highlights were joined. In more detail, low-level data like picture quality, and significant level highlights such as movement arrangement and face acknowledgment to identify consideration districts of a picture were used. Results showed that their model delivered a good performance using MSCOCO, Flickr 30K, PASCL, and SBU datasets.

In the proposed model of [3], the integration of input(image) captions produced by the Fully Convolutional Network (FCN) and the age of picture inscription was investigated. This method was presented as a solution to the problems of missing image data and deviation from the image's core substance.

The authors of [4] investigated several transfer learning approaches to increase the accuracy of image captioning systems. Here, a variety of state-of-the-art models were used to generate the feature vectors, which is the input of the Encoder-Decoder Network. This network is based on stacked LSTMs with soft attention and embedded. The studied models were compared using several benchmarks and using multiple evaluation metrics such as BLEU and METEOR.

In [5], both deep learning convolutional neural networks (CNN) and recurrent neural networks were used to construct well-defined and relevant captions. CNN was used to extract the features of the images. The authors then used the extracted features to create a language model for individual word units using long-term short-term memory (LSTM). Finally, the obtained image representation which also called as fingerprint is obtained using a soft-max activation function.

The authors of [6] demonstrated that image caption approach requires extracting image contents using computer vision, and NLP to obtain the appropriate caption, where they suggested to use the Convolutional Neural Network (CNN) is a powerful image extraction tool, while Gated Recurrent Unit (GRU) is used for efficient caption creation. Overall, the developed model obtained a higher BLEU-4 score on the MS-COCO 2017 dataset when compared to other works.

A new time-varying parallel RNN (TVPRNN) was proposed in [7] to deal with the challenges of caption generating. TVPRNN used two common CNNs (inception v3 and VggNet) to extract global image features, as well as RNN to extract time-varying features at each sample of time, which were then used to represent current words. In a multimodal space, textual and visual representations were combined. The authors evaluated the approach using the datasets Flickr8k, Flickr30k and MSCOCO datasets, and the resulting outputs showed that TVPRNN outperformed the state-of-the-art methods.

The authors of [8] tackled the issue of automatically creating captions for an image based on its annotation. Based on [8], most of the previous studies used computer vision techniques to first determine the labels, then used the existing descriptions of the images in the training subset to generate new captions for the tested image. Also, they claimed that none of the existing studies has reported results about the impact of wrong label on the overall performance. The authors used the PASCAL sentence dataset. They worked

on the level, defining the many types of word in the description such as subject, object, preposition, verb and so on. Various features of the constituents , such as : the verb's tense (present) and aspect (progressive), the verb's form (active or passive), the noun's form (plural or singular), the attribute's position (prenominal or post nominal). They created new descriptions based on the current captions, which deal with the properties of words in sentences.

The authors in [9] work on producing an automatic image annotation was a high performance to be used and integrated in image searching system. As stated in [9] if annotation tools are not used, the searching system may rely on human descriptions or the enormous volume of text on the surrendering area of the images. The authors ran into an issue where users could provide insufficient or noisy tags, and all the text on a web page might not be a description or related to the input image. Therefore, the authors developed an effective tool to automatically annotate images with accurate and sufficient tags. Along with the content of the image, the context of the image plays an important role in this process.

The study in [10] proposed a framework that integrated a series of heuristic patterns for the feature extraction. Their experiments and results showed that the proposed system achieved an 85% as average precision, a 73% as recall, and the average F-measure was 0.78. According to the results, the proposed technique provides effective outcomes for the feature and aspect product.

The authors in [11] presented a system that automatically generates natural language descriptions from images that exploits both statistics gleaned from parsing large quantities of text data and recognition algorithms from computer vision. They depended on image labels that were gained by after using segmentation of the images. Figure 1.1, shows the input image without segmentation, and Figure 1.2 shows the output(captions) for the same image using segmentation algorithm.

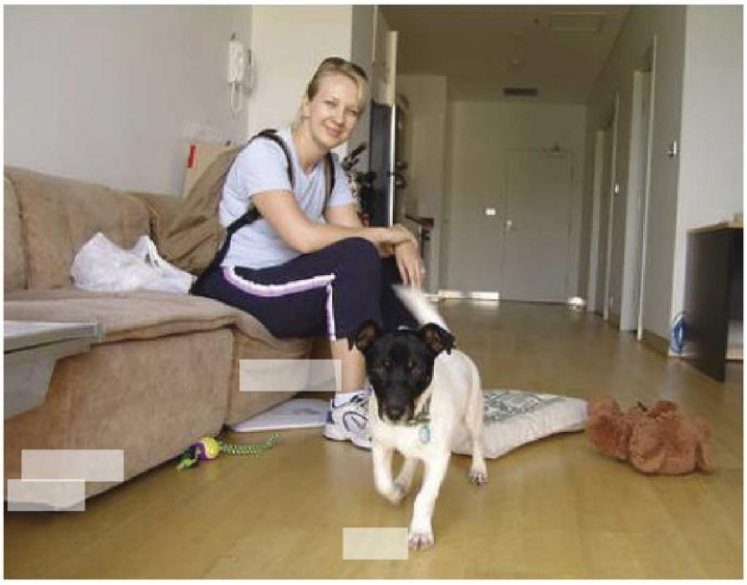

Figure 1.1: An image example before segmentation [12].

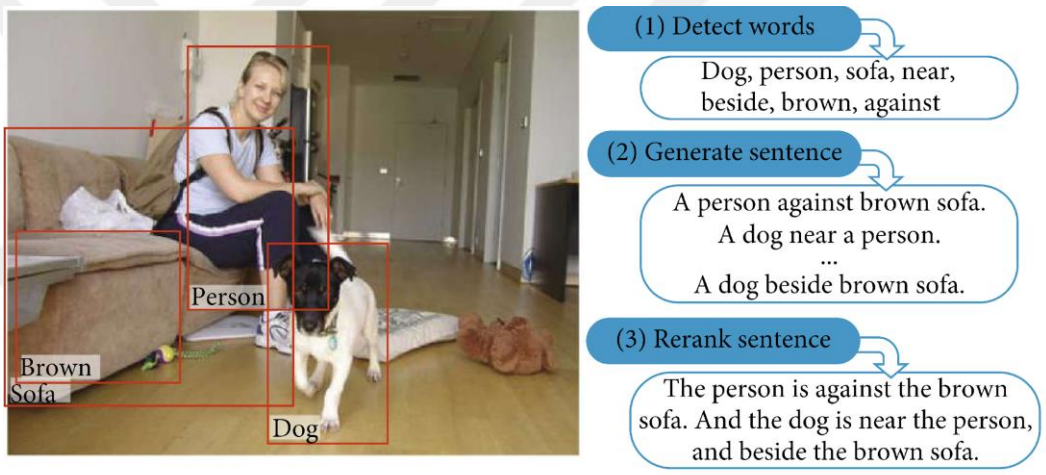

Figure 1.2. Labeling image using segmentation [12].

1.5.1 **Our Work vs State-Of-The-Art**

We aim to develop and enhance a new system that describes images by some sentences and/or assigns a caption for each image. Based on our preliminary investigation, our developed system depends on VGG19 architecture as we believe it's more suitable compared to other models such as AlexNet and Googlenet. A brief comparison for VGG19 and AlexNet which were the most suitable two models is shown in Table 1.1.

Table 1.1: A brief comparison between VGG19 and AlexNet.

AlexNet	**VGG19**
The first layer has a filter with kernel size 11x11 and the second layer has a 5x5 filter.	In VGG19 all the convolution kernels are of size 3x3.
There is max pooling after every convolutional layer	Max Pooling is done after both the second and the fourth convolution layers.
Using large filter sizes with large strides	VGG19 uses small 3 × 3 filter sizes with stride 1 throughout the whole net
One convolution with a kernel of size 5 × 5	Using the stack of convolution layers (with a kernel of size 3 × 3) incorporates three non-linear rectification layers instead of a single one in 5 × 5 convolution This makes the decision function more discriminative and hence learns more complex features.

In general, the convolution filters with small size enable VGG to enhance its performance. In addition, using multiple small filters (In VGG19) will not cause losing any information. Moreover, using sequences of small filters makes the decision function more discriminative and learn more complex features. This allows the model to create a better mapping each image to the most suitable label. Also, as the parameters decrease using the stack of 3 × 3 convolution layers. Assuming both the input and output have C channels, the stack of 3 × 3 is parametrized by $3\times(3^2C^2) = 27C^2$ weights whereas (in some cases of AlexNet) 7 × 7 will require $1\times(7^2C^2) = 49C^2$ weights, which is 81% more. Hence, using 3 × 3 Conv layers decreases the size of the model on memory and also acts as a sort of regularization, which makes the network less prone to overfitting. Therefore, VGG19 is preferred in our research. It is worth mentioning that the above-mentioned information does not illustrate that VGG19 can always outperform AlexNet nor vice versa. The decision of which model is better is depending on the case study.

The flowchart of the developed approach is shown in Figure 1.3. Briefly, as shown in Figure 1.3, the developed system has two main steps:

- Computer vision step which is done mainly using deep learning model, VGG19 in particular, and works on extracting efficient feature maps that represent the input image.

- NLP step that applies text preprocessing techniques on the caption and/or description of each input image.

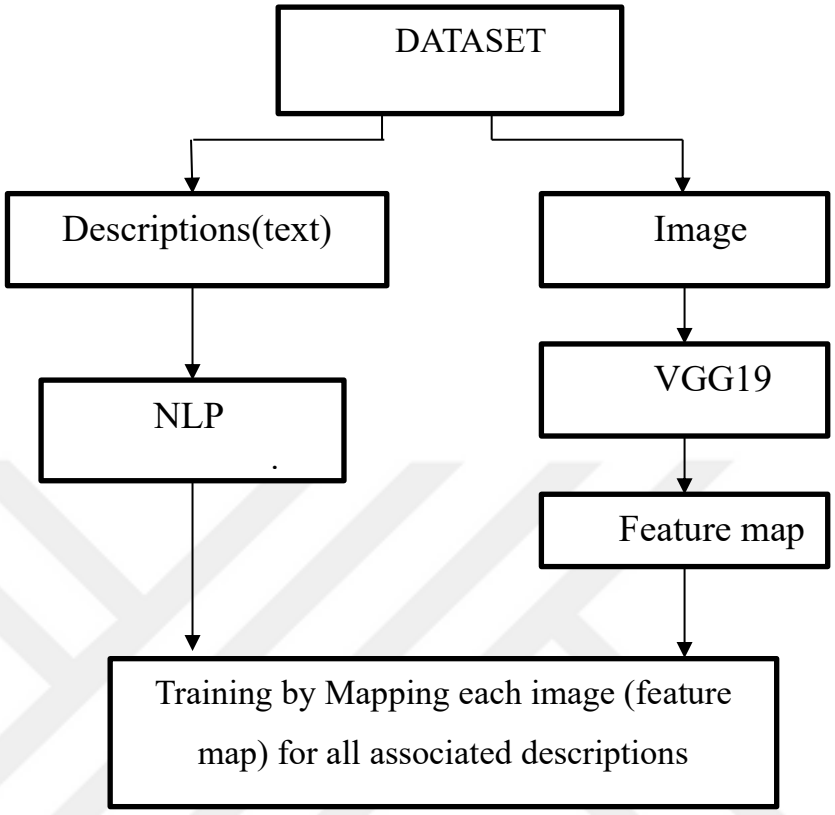

Figure 1.3. Flowchart of traning the developed approach.

Our final developed system should be able to generate captions for each input image(Figure 1.4 and Figure 1.5). The main steps to obtain the caption are: 1) Preprocess the input image (some dimensions edits). 2) Use the enhanced VGG19 to get the feature maps. 3) Finally, LSTM will use the feature maps to generate the most suitable caption.

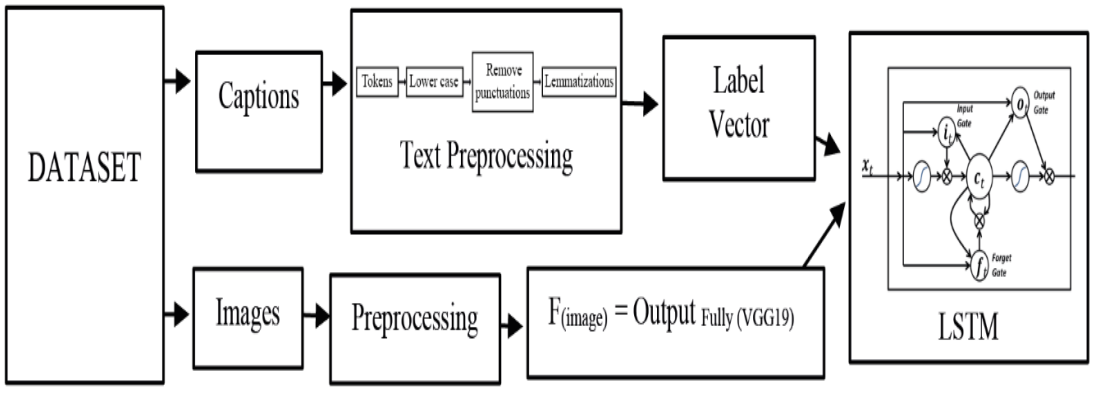

Figure 1.4. The main steps to obtaining our trained model.

Where F(image) stands for the image's features, Output(Fully (VGG19)) is the output of the fully connected layer , which is the last layer of the used VGG19.

It is worth noting that the used LSTM was improved by 1) using genetic algorithms to determine the best batch size, 2) investigating multiple optimizers and selecting Adaptive Moment Estimation (Adam) as the best optimizer, 3) adding a dropout layer to keep the loss function as small as possible, and finally 4) adding a regularization term to avoid overfitting.

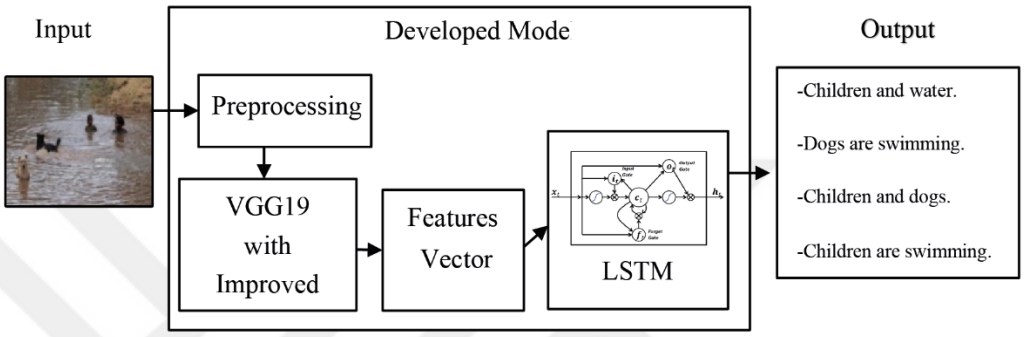

Figure 1.5. Main steps of assigning a caption(s) for the input image using our developed approach.

1.6 Structure Of The Thesis

This thesis is divided into five chapters. The background of this research including the main problems, objectives, and literature review are described in chapter 1. In chapter 2, Image Annotation and the related Image Annotation and Natural Language Processing methods are provided. Chapter 3 illustrates the implemented system. Chapter 4 present the discussion of the experimental investigation of the developed method. Conclusions and future work are stated in chapter 5.

CHAPTER 2

CONCEPTS AND ALGORITHMS FOR IMAGE ANNOTATION

Assigning captions by people is an easy task, as the human brain has the ability to analyze images, understand the content and decide the appropriate caption. However, it is much harder to automatically create a caption, by some device or software, using most of the existing methods and algorithms. Hence, some advanced methods in language processing and machine learning are required to build an efficient system. This chapter presents the details of the needed and related tools and algorithms to achieve the required task. We begin with natural language processing. Then, we explained the deep learning, image encoding and decoding, and finally text processing is presented.

2.1 Image Annotation and Captioning

As mentioned before, providing short text captions for an image is called captioning or image annotation. This is an important problem that we need to be improved and automated [13]. In other words, creating short captions by looking at an image is easy because people have the ability to easily understand and combine the objects of an image and the interactions between these objects to create meaningful sentences. These human abilities have been accumulated through long observation and teaching. Therefore, it is not easy to reproduce these behaviors on a machine and expect human-level accuracy. One possible direction is to train computers to learn from various images, gain experience, and understand relationships between objects so that they can generate meaningful descriptions in a manner similar to human experiences.

Image annotation in general needs to create image dictionaries, i.e., bag of visual words that will be used to train and build such application. Figure 2.1 shows an example of image dictionary and its common words that can be used for preparing image annotation system.

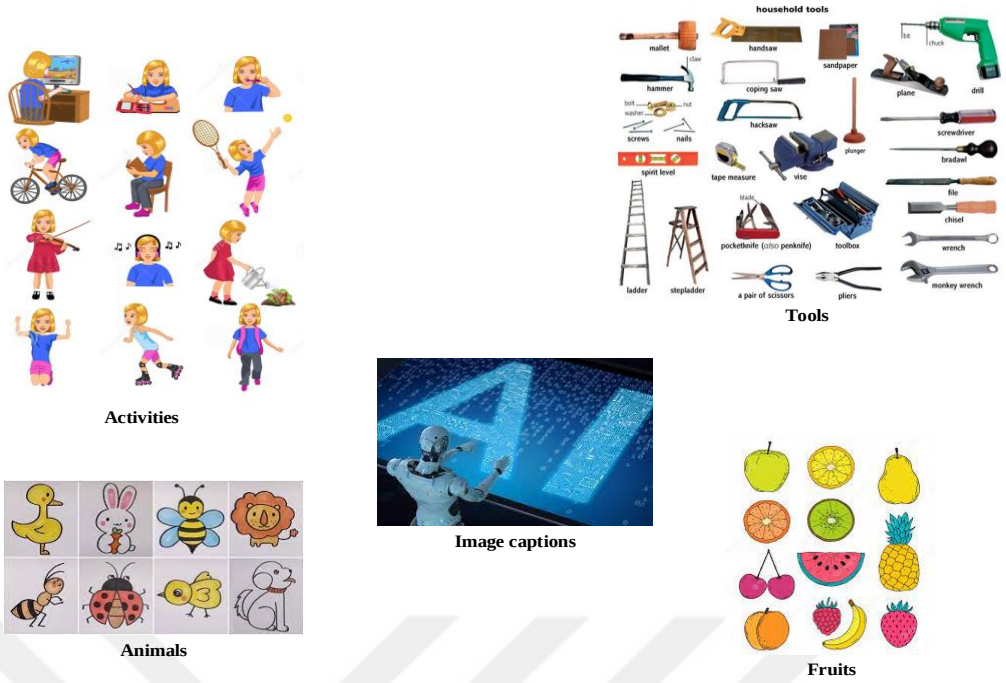

Figure 2.1. A sample dictionary that can be used for image captioning, Where all the images in each subgroup are related to each other. For example, a subgroup is used for each of fruits, tools, toys, etc.

It is worth mentioning that each image may many descriptions, For example, the image shown in Figure 2.2 might be described as animals are playing in the snow, or as two small bears are running.

Figure 2.2: A sample image input that might be described by multiple statements.

Since captions are words in a specific language; in our research, we will work on

generating the English captions. Natural Language Processing is a common term in the text processing field, which is discussed and explained briefly below.

2.2 Natural Language Processing

Natural language processing (NLP) is an artificial intelligence, computer science and linguistics subfield. It focuses on how humans and machines interact with each other. How to educate computers to represent, manage, analyze and understand text data in particular. The desired goal is to build a system that can accurately extract and retrieve meaningful information from text documents. Additionally, the device has the ability to arrange and categorize the documents.

As shown in Figure 2.3, NLP uses three main techniques: syntactic, semantic and pragmatic. In the syntactic technique, the natural language is analyzed with the rules of a formal grammar. Whereas the semantic technique is the process of understanding the meaning and interpretation of words, signs and sentence structure. As for the pragmatic technique, it is the part that extracts information from texts. It is the branch of NLP concerned with determining the true meaning of a set of text structures [14].

The employment of approaches in NLP has changed over time, as seen in Figure 2.3. As it is predicted, NLP will solely be based on pragmatic techniques by the year 2100. The pragmatic technique might be widely employed to substitute people in multimedia applications.

In our proposed approach, NLP uses syntactic and semantic techniques. The goal is to create captions that describe communication, activities, meaning or emotions. To do so, image and text processing technologies are required. Deep learning is one of the famous tools for handling images and texts ; therefore, it will be used in our work. The next section presents an explanation of deep learning.

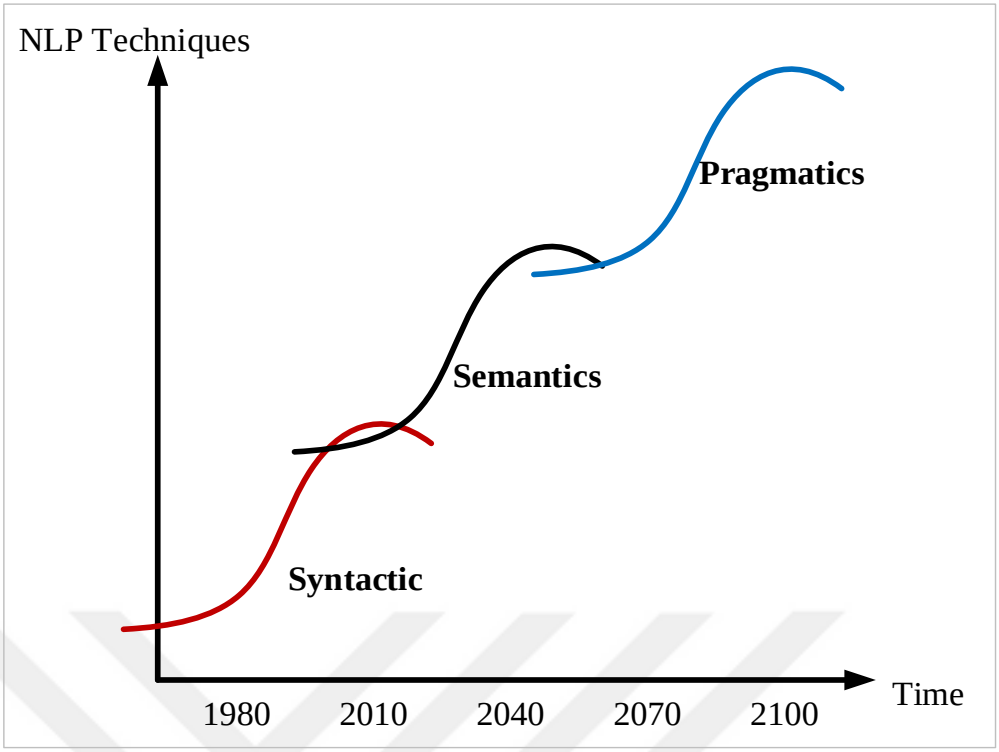

Figure 2.3. Evolution of NLP research.

2.3 **Deep Learning**

As illustrated in Figure 2.4, deep learning is a subfield of machine learning in general. Deep learning is also known as deep neural networks; the name comes from the architecture of the system; deep neural networks resemble a neural network with deep learning features.

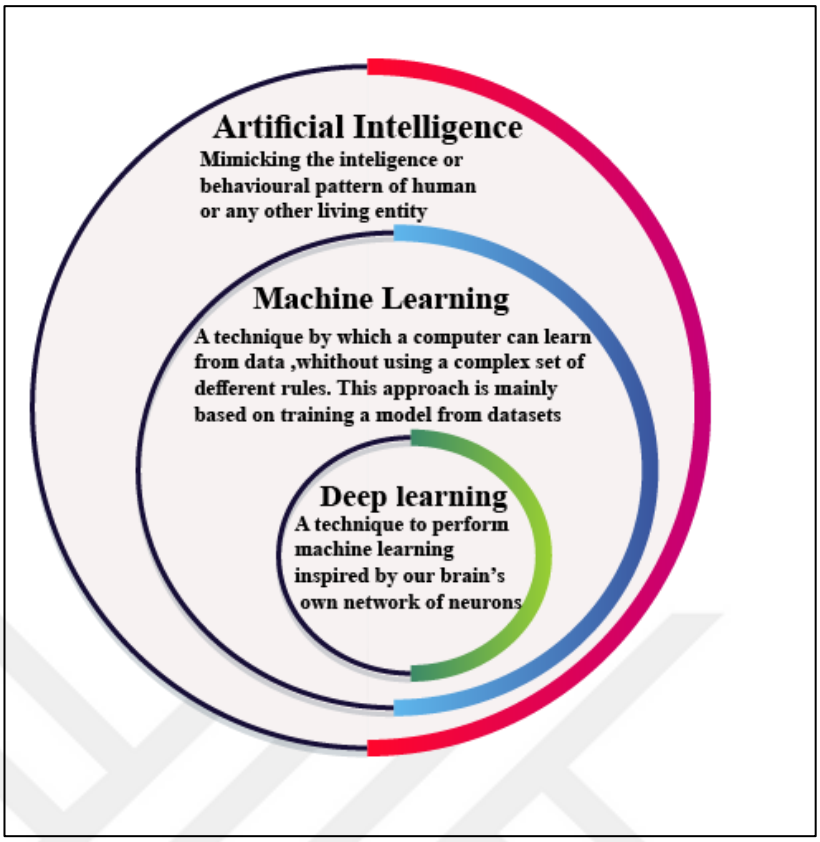

Figure 2.4. Deep learning and machine learning.

The concept of work in neural networks is similar to that of human neurons; they are learned from a large amount of data and numerous attempts. Available data will have distinct characteristics that set it apart from other data. To create the desired result, a deep neural network contains an input layer (which handles the input (in most cases, the input is an image or an array of data), hidden layers (many hidden layers), and an output layer. The number of hidden layers of a network is used to determine its depth.

Following a general overview of the fundamentals, general techniques for text and image processing will be discussed in the following part.

2.4 General Techniques

A captioning tool for images includes an image feature extraction tool (called image encoder) as well as a language model. The feature extraction tool converts an image into a feature vector (called feature map). Deep learning techniques, such as CNN, are used in this tool. The linguistic model generates a caption in a sequential manner.

There's also an image decoder for retrieving the image's caption. Encoding and decoding

examples are shown in Figure 2.5.

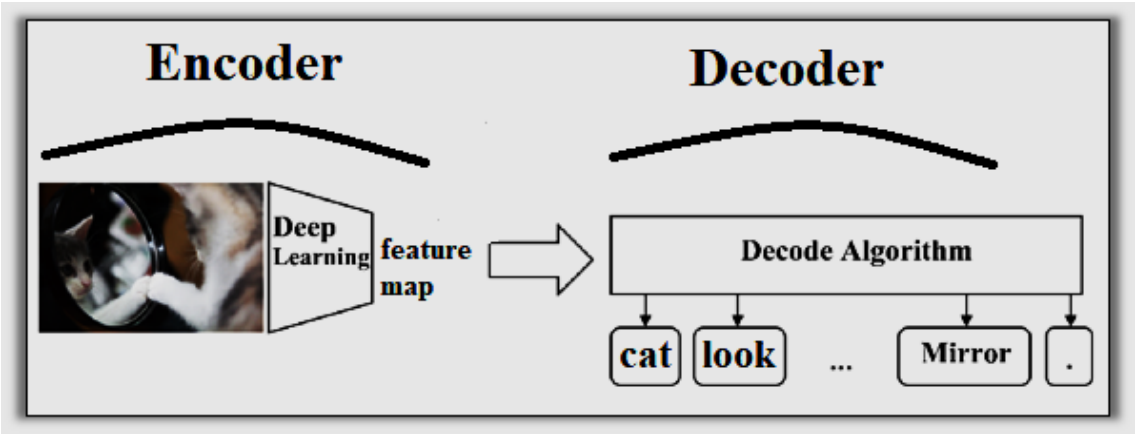

Figure 2.5. The structure of encoder and decoder for an image captioning system.

In many studies, a CNN model was used to encode images, while Recurrent Neural Network (RNN) is frequently utilized to process text. Long-Short-Term Memory (LSTM) cells or Gated Recurrent Units (GRUs) are commonly used to create text models since they are better at memorizing longer sequences. In NN, ENCODER and DECODER are implemented in multiple ways and approaches based on certain rules, such as extracting features or forming attributes without looking at their type as words. However, we are lucky as currently many large databases are available for training an efficient system.

The following are the main parts of our developed approach:

- Image encoder to extract features from images; it is VGG19 which is a deep learning model that depends on CNN.
- Image decoder to build captions which is an LSTM algorithm.

In the following paragraphs, we will go through some techniques that are used in image caption and description.

2.5 Image Encoding

Considering an encoder of feature extraction as a black box, its input is a raw image with dimensions 3×W(Width)×H(Height) with no pre-processing steps. It also produces a C×W1 ×H1 tensor called a feature vector of a feature map. A feature map may be handled by a flatten layer to be a feature vector, where C,W1, H1 is the dimensions determined

and produced by the used layers that are utilized in decoding. The kernel size of the convolution and pooling layers in the deep learning algorithm used in decoding determines W and H.

In general, the encoder is a CNN algorithm that has been trained on a variety of tasks related to a large number of image datasets, such as image classification. When a certain CNN is chosen, the top layers may be omitted in some circumstances [15], and the last layer is used to get the image label.

2.5.1 **CNN**

According to [16 and 17], Convolutional Neural Network (CNN) is a Deep Learning algorithm, where its input is an image, and it has connections between the neurons with changeable weights (learnable weights). CNN assigns weights to various aspects and objects in the input image. It can differentiate one image from another based on these aspects. In comparison to other classification algorithms, convolution layers in CNN are less sensitive to process raw images(without pre-processing). Other classification algorithms use image pre-processing techniques that are manually engineered to find image features. Figure 2.6 illustrates the general design of CNN.

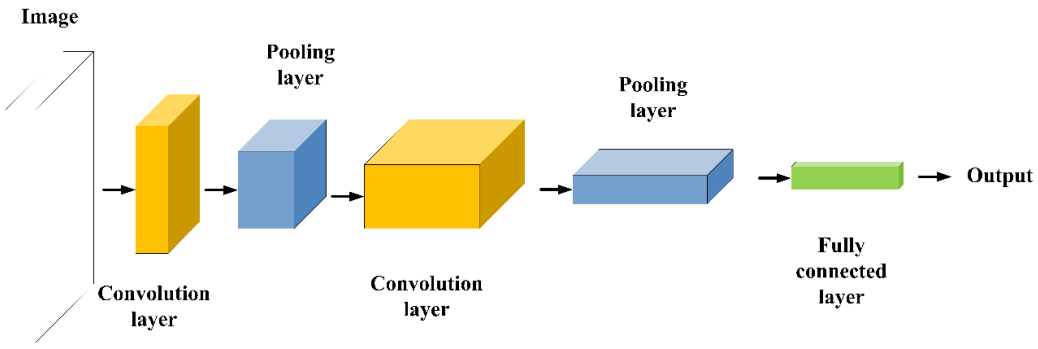

Figure 2.6. General Desing of CNN

A convolution layer, a pooling layer, and a fully linked layer are the most frequently used layers in most of the CNN models. In general, a dot product between two matrices is computed in the convolution layer. The first matrix is the kernel one that contains a set of learnable parameters. Whereas, the second matrix is the restricted portion of the receptive field, where the Receptive Field (RF) is defined as the size of the region in the input that produces the features. To depict the receptive region, the kernel slides over the image

(height and width). As a result, an activation map, i.e., representation of the image is created, which depicts the kernel's response at each spatial position of the image. It is worth mentioning that the sliding size of the kernel is called a stride. On the other hand, the function of the pooling layer is to gradually lower the spatial size of representation by calculating a summary statistic of surrounding outputs. This operation is performed separately on each slice of the representation. Many functions, such as the max pooling, L2 norm, and weighted average, can be utilized as pooling functions.

Another type of layer is the fully connected, and as the name indicates, its function is to map the representation between the input and the output. Neurons in a fully connected layer have full connectivity to all activities in the preceding and succeeding layers. Moreover, there is also a layer named "dropout" that is responsible for ignoring some neurons during the training stage by a constant named “Keep Probability” [18].

2.5.2 **VGG19 model**

VGG-19 is a well-known example of a 19-layer convolutional neural network. Here, we will use a pre-trained version of it, which was trained using a million photos from the ImageNet database. The pre-trained VGG19 can classify 1000 different objects, including mouse, mountain, keyboard, and many others. It is worth mentioning that the VGG-19 is a better version of the VGG-16 and is created by combining convolutional layers. However, the depth of the model is limited, which makes deep convolutional networks difficult to train. Figure 2.7 shows the dimensions of VGG19 layers.

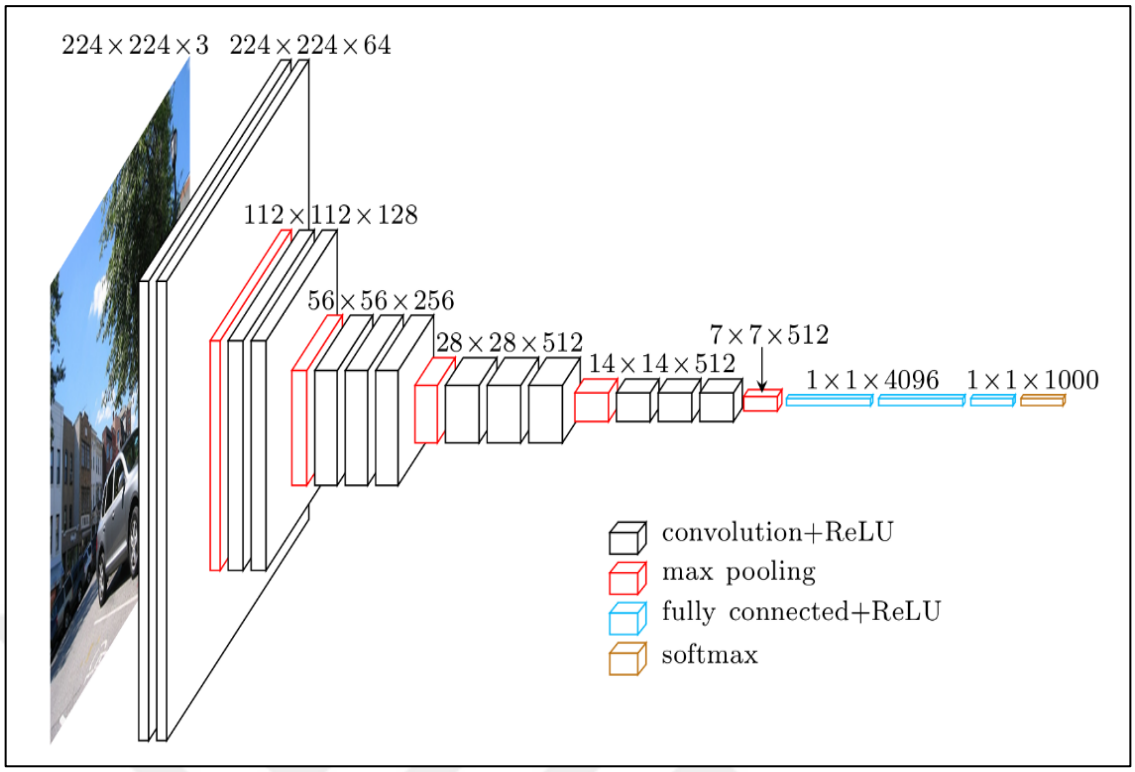

Figure 2.7. Image and features dimensions for layers of VGG19 [19].

2.6 **Image Decoding**

The image decoder describes how an image feature vector is converted into a set of tokens(words) or single word. In general, experts create templates by selecting the most likely objects, actions, qualities, and prepositions based on models like the Hidden Markov Model. By their structure, templates strongly determine the structure of the generated sentences.

A language model is a probabilistic model that returns a probability for each word in a defined vocabulary of being the next word given a set of words (in the field, the unit of word is defined as gram). Traditionally, this can be done using word n-grams or Neural Networks (NNs). A word n-gram is a sequence of any set of words that appear in a certain text. NNs are continuous language models, while word n-gram models are discrete. Nowadays, Long Short-Term Memory (LSTM) is very frequently used as a decoder to predict captions. LSTM is a variant of Recurrent Neural Networks which are usually used with sequential data such as text and audio [20]. When features are fed to LSTM, it depends on the impersonation of all available information and give a phrase. Long Short Term Memory (LSTM) is the decoder used in our method.

2.6.1 **LSTM**

The main part of the LSTM model is the memory cell, as shown in Figure 2.8, referred to it by "C", which stores all information about image features and previously generated words, and keeps track of the functions of all three gates [21], which are:

- The input gate takes words from the memory cell, combines them with the last input, and then stores them in the memory cell.
- The forget gate takes words from the memory cell and decide to either forget them or keep them according to the input.
- The output gate combines words (from input or from the memory cell) with the last output (in time $(t-1)$).

  Note that the input is handled by the three gates. They are integrated in order to preserve the last input.

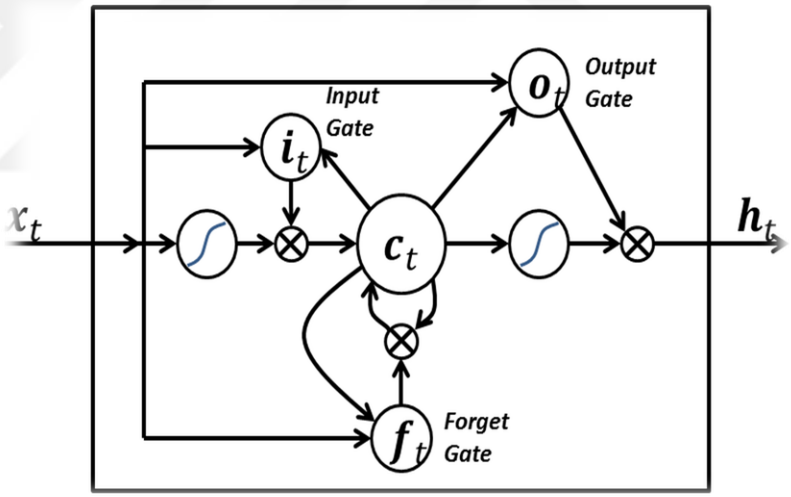

Figure 2.8. LSTM Structure [22].

2.7 **Text Processing**

Text preprocessing is a basic stage in any text-based project. Human languages (of various types) contain text data that can be measured in units such as words. These words have different structure, meaning and use; they might be emojis, special symbols, or short words, etc. It must be cleaned of noisy units before being used by the machine-learning model.

Preprocessing text data requires a variety of different steps. It is worth mentioning that when captions are generated for input images, multiple ones can be generated. For example, the image, shown in Figure 2.9, has five different captions, i.e., ['A pink-dressed child is climbing upstairs in an entryway.', 'A girl is going into a wooden building.', 'A little girl is climbing into a wooden playhouse.', 'A little girl is climbing the stairs to her playhouse.', 'A little girl in a pink dress is going into a wooden cabin.']

Figure 2.9: An image example that has multiple captions [23].

The main preprocessing steps that we integrated in the developed system are explained below.

2.7.1 **Tokenization**

Tokenization is the first well-known NLP process. Both modern Deep Learning-based architectures and traditional NLP methods rely on it. Tokenization is also the process of analyzing and breaking down a paragraph of text into smaller units known as tokens. Sub-words, words, and characters can all be used as tokens. Take the following sentence as an example: "Never give up." Splitting tokens with spaces is a well-known method of obtaining tokens. Sentence tokens will be 3 words, i.e., Never, give, and up. It's called Word tokenization because each token is a word.

Consider the word “smarter”:

- Character tokens: s, m, a, r, t, e, r
- Sub-word tokens: smart, er

2.7.2 **Lower Captions:**

If this pre-processing step is skipped, the words NLP, nlp, and Nlp will all be treated differently. With this process, words will be converted to the same form, i.e, one word.

2.7.3 **Remove Punctuations**

The evaluation criteria take into account punctuation marks, which clearly affect the improvement of the performance; for instance, the word "Quickly" will be indexed, and also the word "Quickly!" is indexed as a different word. Therefore, punctuation marks must be removed. Also, in some cases, the word embedding model may not support embedding the punctuation marks and therefore will give different results.

Note that other processes steps such as the following might be integrated and used:

- Remove characters like 'the, a, an'.
- Keep only the alphabets (remove numbers).
- Remove words that count less than a specific threshold.

2.7.4 **Lemmatization**

The process by which several different forms of the same word are mapped into one single form, which can be called a root form or base form, is known as lexical notation (Lemmatization) in NLP. It saves data space and eliminates the need to check every form

of a word by reducing the number of forms a word can take. This allows us to overlook morphological changes within a single word.

Consider the following example: we have a set of documents, and we are trying to retrieve all documents related to the word "dance". We need to look up the words "dance," "dancing," "danced,", etc. Alternatively, we can reduce all forms of each word to a single one, in our example into "dance". Headings can be done while storing or retrieving data. Ideally, we should keep both the original form and lemma, and use them as needed. Most NLP projects, as well as any type of machine learning project that deals with words, can benefit from using headwords. This is advantageous because it aids in the normalization of words and the reduction of space.

2.7.5 **Frequency**

The word frequency can be represented by multiple ways, for example the frequency of each token in the document, or using some advanced techniques such as using unsupervised learning algorithm to generate word embedding by aggregating global word-word co-occurrence matrix from the used dataset(corpus). This embedding technique is effective and frequently used. In addition, it should also distinguish between tokens and types. Tokens are words including repeats, whereas types are the distinct words in the corpus.

In other words, the word embedding is a way to represent sentences and words using a dense vector representation [24]. GloVe, Word2vec, and ELMo are example of word embedding approaches and their details as summarized below.

GloVe stands for global vectors for word representation and is a well-known example of word embedding [25, 26, 27]. GloVe focuses on word co-occurrences across the whole corpus. Its embedding relates to the probabilities of two words appearing together.

Word2vec is another word embedding technique. Word2vec is a two-layer neural net that processes text by "vectorizing" words. Word2vec models' vector representations of words have been shown to carry semantic meaning and to be useful in a variety of natural language processing (NLP) tasks. The goal of word2vec is to group similar word vectors together in a workspace. Using these vectors, word2vec can make highly accurate guesses about the word’s meaning based on past appearances [28].

ELMo is a new word embedding technique that creates word representations using a deep bidirectional LSTM model. ElMo analyzes words in the context in which they are used, resolving the problem that the context of a word has a significant impact on its meaning. The bi-LSTM is made up of many layers, each of which contains different information about the context, which is combined in order to improve the output, i.e., test's features or representative [29].

CHAPTER 3

THE DEVELOPED SYSTEM

Our method is based on the VGG19 model, which will be discussed in detail in the following paragraphs. Feature extraction (or image encoding) and text processing are the two phases of the process (Image decoding), which followed by training and validation phases. Figure 3.1 illustrates the entire process of our developed system.

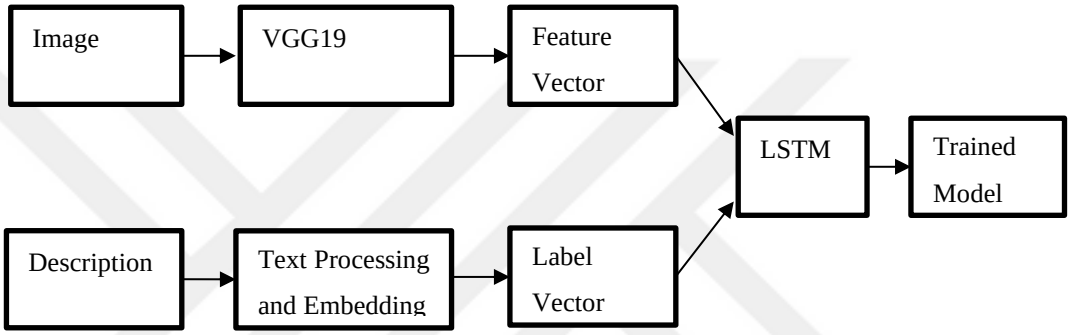

Figure 3.1. The main structure and components of the developed system.

VGG19 is used to extract feature vectors or feature maps for each input image. Text preprocessing and embedding are used to convert captions to label vectors. Both feature map and label vector will be used as input to train the used LSTM.

3.1 Feature Extraction

As previously stated, feature extraction or image encoding is one of the main steps in our deep learning-based system.

3.1.1 VGG19 and Enhanced VGG19

The designed VGG model is illustrated in the Figure 3.2.

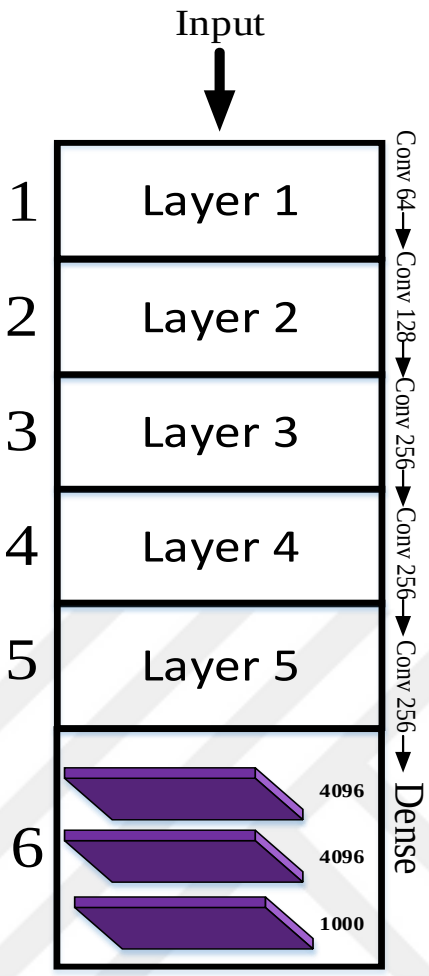

Figure 3.2: The structure of the used VGG model.

shown in Figure 3.2, layers 1 and 2 consist of two convolution layers followed by a pooling layer, whereas layers 3, 4, and 5 have four convolution layers followed by a pooling layer. Where in the case of enhanced VGG19, two changes have been made: 1) Adding the Regularization term to convolutions: Regularization enables us to achieve penalties on parameters in CNN layers during optimization. These penalties are combined with a loss function that the network optimizes. The most well-known regularization types are L1 and L2. By summing other terms related to the Regularization term, these functions update the existing and used cost function.

$$Cost\ function = Loss\ (ex: binary\ cross\ entropy) + Regularization\ term$$

This regularization term, however, is different in L1 and L2. Regularization is a useful tool for penalizing composite models in machine learning; it basically reduces overfitting by setting network weights to tiny values. Furthermore, it improves model accuracy.

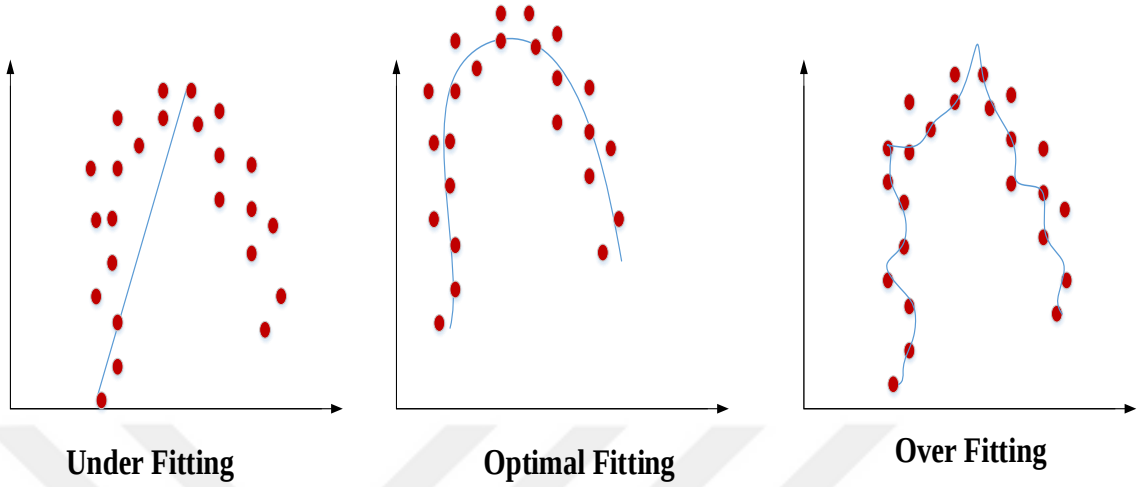

Figure 3.3. Shows an example of the over- and under-fitting.

2) Adding a dropout layer to fully connected layers: In classification tasks, a dropout layer is useful for dropping out some neurons in fully connected layers during the training phase. Dropout will delete some neurons in a feature extraction task to reduce the loss function as much as possible.

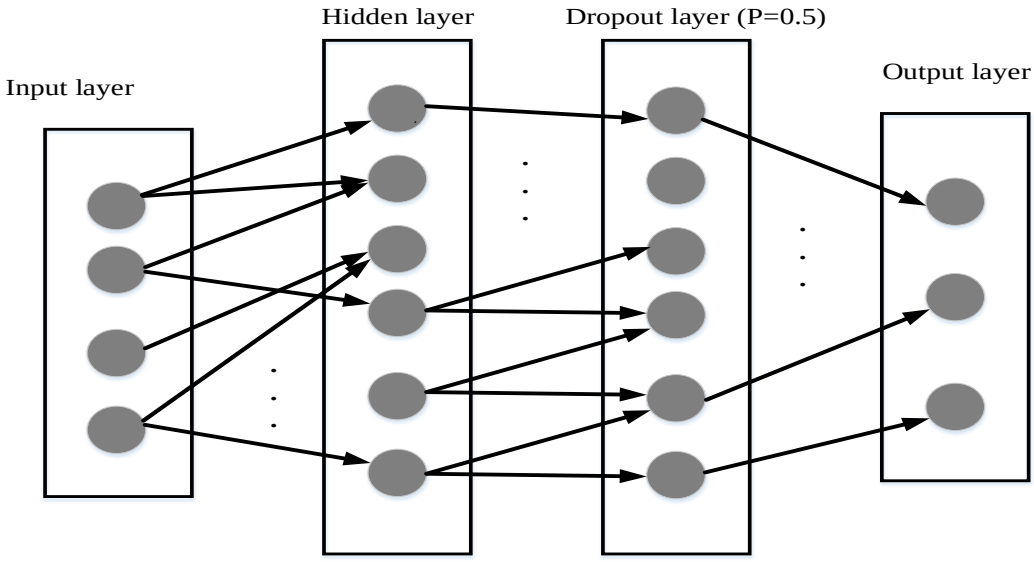

Figure 3.4. The dropout layer drops neurons from the final layer.

Between the LSTM layers, a dropout technique is used. Every token or word's meaning as well as its hidden states and vectors are saved during the embedding phase. This means

that if a word is used after a certain number of words (say 50) in a text, the calculations must be saved in memory, which will result in a very large size for each word. As a result, RNNs are unable to store data in their memory. RNNs are unable to learn these long-term dependencies for this reason. On the other hand, LSTMs work well with this type of text. Also, LSTM networks work well with time-series data.

3.2 **Developed System Design**

Prepare dataset, text preprocessing, words embedding, dataset splitting, features extraction, training phase, and test model are the six steps in the system design. Each step is explained in detail in the following paragraphs.

3.2.1 **First Step: Selecting Dataset**

The Flicker8K dataset is a well-known dataset that has been used in many studies, and it has been used to train and test our proposed model. Following the evaluation, our proposed model can be tested on other datasets, such as the flicker30k dataset. There are five labels for each image in the used dataset (captions or descriptions). Flicker8K can be downloaded and used for free. It contains about 8092 JPEG images of various dimensions. It is split into three parts:

- Training part (6000 images)
- Testing part (1000 images)
- Development part (1000 images).

Also, the dataset contain txt files that has labels for images. For each image, there are five captions (total 40000 captions). Table 3.1 shows a sample of the dataset images.

Table 3.1. Shows a sample of the data used, demonstrating that there are multiple captions for the same image.

Image	Caption
	Game of basketball
	Three boys in woods
	Boys playing in street

3.2.2 **Second Step: Captions (Text) Preprocessing**

The captions are handled and processed in this step (as described in Text Processing). The Figure below illustrates the entire process.

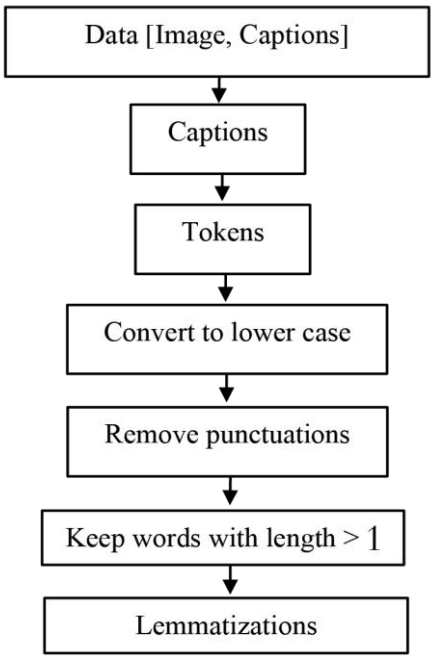

Figure 3.5. Illustrates the steps of text preprocessing.

Following text processing (as shown in Figure 3.5), there is structure data where the first column is image data and the second column is all captions for the images. All captions are also counted for the number of vocabularies.

3.2.3 **Third Step: Word Indexing and Dataset Splitting**

In this step, each input is represented by a numerical vector using the used word embedding algorithm, i.e., which as shown the experimental results obtained the best performance and outperformed the studied algorithms.

For training and validation, the dataset is divided into two parts. The model is trained with the dataset's training subset and then validated by comparing the results using the validation subset. Typically, 80 percent of the dataset is used for training and 20% is used for verification. According to this, 80% of the data (text, images, labels) is used to train the model, while the remaining 20% is used to validate it.

3.2.4 **Fourth Step: Features Extraction**

To create a feature map, the used VGG19 model, shown in Figure 3.2, is used to extract images features. In this step, the image input will be represented as a feature vector with dimensions of (1*1000). The details of this step are explained in the previous paragraphs when discussing CNN and VGG19.

It is worth noting that the used enhanced VGG19, we have further works on improving the model by tuning the following parameters.

1. **Batch Size**

For deep learning models like CNN and other machine learning models, the batch size is an important and effective hyperparameter. We must provide the data in batches for ML or DL models to be able to find representative features. By using a batch size (for example, 10 images at a time), the model is able to distinguish between features by handling all the batch's samples. Given that we have a model that we want to train to classify images, we can do this by supplying one image at a time (batch size equals one), or we can provide 10 input images at a time to the model, and the model will be able to distinguish common patterns like head, arms, and legs based on these 10 images.

Genetic Algorithm for Optimal Batch Size

Various batch size values are used on researches like 32, 64, 128. However, we recommend using Genetic Algorithm to optimize the batch size value in order to improve accuracy and results.

According to [30], Genetic Algorithm (GA) is a mathematical algorithm for determining a specific parameter that minimizes the loss function. Genetic Algorithm uses the evolutionary generational cycle to produce high-quality solutions. See [31] for more information on GA. In general, GA process starts by searching for a range of values to find the optimum solution or a solution that is close to the optimum. It generates random solutions called "initial solutions" (initial population), with each solution represented by a chromosome. The chromosome is a group of solutions, each of which has a fitness value calculated by the fitness function. The chromosome with the optimal fitness value is chosen as the parent, and the remaining chromosomes are fed to a genetic operator that performs crossover to generate subsequent generations.

In this work, the GA's task is to find the optimized batch size that minimizes the loss function over a range of values for the batch size. Figure 3.6 shows how GA is used for batch-size optimization. The parameter to be optimized in Figure 3.6 is X (batch size). The fitness function is also known as the loss function. On the parameter X, there are two constraints: X is a number that is less than 256 and divisible by eight (it is preferable that the batch size is divisible by 8). Y is the loss function value that corresponds to the optimized value Xopt. After applying GA, we found that Xopt=64.

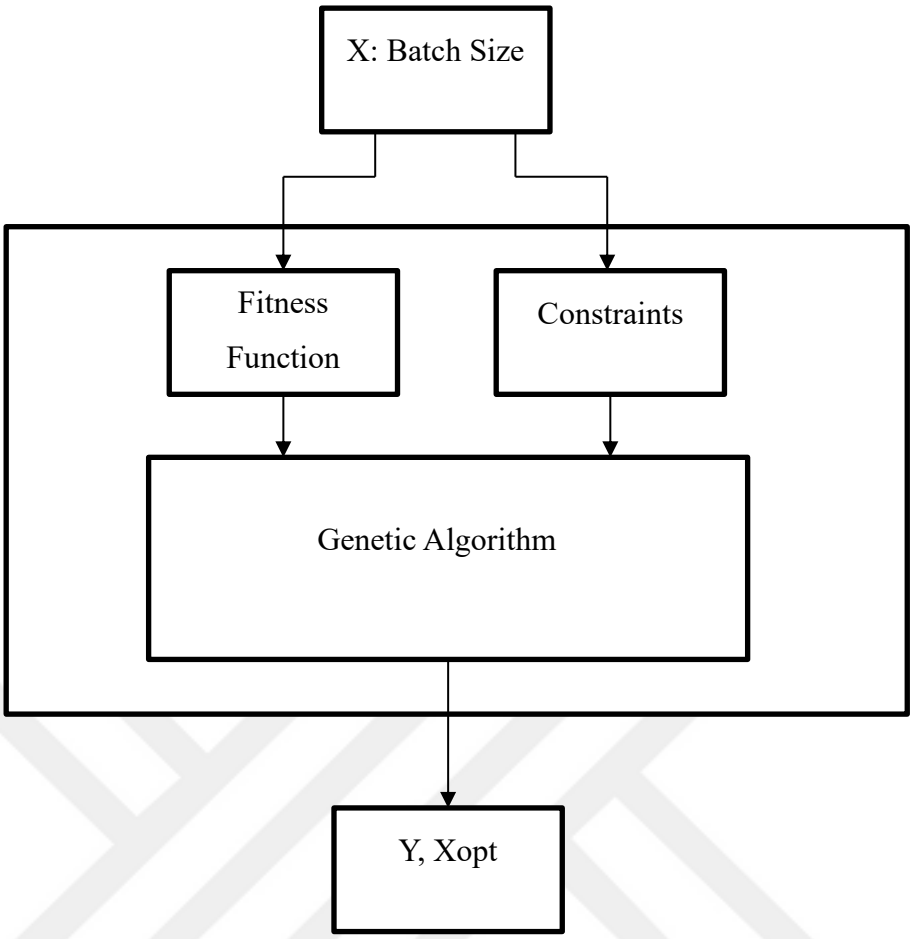

Figure 3.6. Optimizing batch size using Genetic Algorithm.

2. Optimizer

The optimizer works on improving performance and learning speed, the right optimizer is essential for deep learning models. Also, loss function is simply a method of calculating how well a learning model performs. Hence, hyperparameter tuning and weights tuning are done during the training of the ML model to minimize loss and try to make prediction accuracy as accurate as possible.

In other words, optimizers are methods for reducing losses in machine/deep learning models by modifying attributes such as learning rate and weights. Below are some examples of optimizers.

a) When minimizing the cost function in training, the Adaptive Moment Estimation (Adam) optimizer performs better (it reaches a global minimum faster and more reliably).

b) SGD(Stochastic Gradient Descent): tries to find minimum or maximum error via iteration. Drawback: one of its major flaws is that when the objective function is not convex or pseudo convex, it will almost certainly converge to a local minimum.

c) Nesterov accelerated gradient: with the following example, it is well known that Nesterov Accelerated Gradient is superior. Consider a ball that is rolling down a hill, oblivious to the slope. However, a smarter ball that recognizes where it is going and slows down before the hill slopes up again would be nice. This precision is provided by the Nesterov accelerated gradient for our momentum term.

In addition to the above steps, the following are also selected and used for enhancements: Adding the Regularization term to convolutions, and adding dropout layer to fully connected layers.

To sum up, as the overall enhancement, the following parts have been enhanced as a result of the previous discussion: Regularization, Dropout, Batch size, and Optimizer, as shown in Figure 3.7.

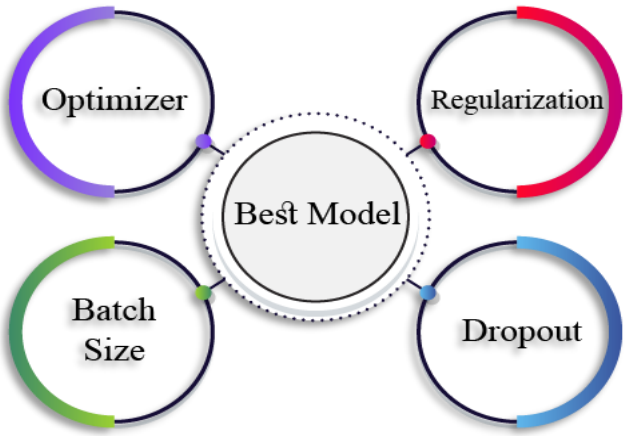

Figure 3.7. Final Enhancment Paramters.

3.2.5 **The Fifth Step: Training Phase**

We have our developed model after completing all of the previous steps. Hence, the following are the model's inputs for training:

- Image feature maps
- Sequence of captions
- The desired categorical output

3.2.6 **Final Step: Testing the Model**

When we have an input image, we can use the trained model to find the most suitable caption. For example, considering the image shown in Figure 3.8 as an input, The model will generate some captions in the order shown in Table 3.2.

Figure 3.8. An example of an image description [23].

Here, the used CNN will generate a feature vector to predict the caption of an image, which will be fed into the LSTM. In the first step, the LSTM predicts the first word W1, which is in our example, 'Girl.' In the first step, S1, which is refer to the output is =W1.

The features vector with the word 'Girl,', i.e., S1, is used in the second step. The LSTM predicts the second word W2, which is 'is,' and adds it to S1, generating S2, where S2=S1+W2, the process continue until we have a complete caption.

Table 3.2. Generating caption in sequences.

Steps#	**Padding sequence**	**Categorical**
1st		Girl
2nd	Girl	is
3rd	Girl is	Running
4th	Girl is running	With
5th	Girl is running with	Two
6th	Girl is running with two	Dogs
7th	Girl is running with two dogs	

CHAPTER 4

EXPERIMENTS AND RESULTS

Multiple experiments to evaluate the developed system are carried out in this section. Three tuning experiments are carried out first. The first experiment is to choose the appropriate word embedding technique. The optimizer that best fits our approach is identified in the second experiment. Finally, the effect of patch size on the performance of the proposed approach is investigated in the third experiment. Next, the proposed approach was investigated and compared to existing models using the BLEU score as a metric. The following sections explain the BLEU metric, and the results of the executed experiments.

B. BLEU SCORE METRIC

To evaluate the proposed system, an evaluation metric that determine how good the generated captions is needed. The Bleu score is a popular metric for evaluating image caption models, and in most NLP projects like language models and summarization, this is a common metric.

The BLEU (Bilingual Evaluation Understudy Score) calculates the number of consecutive words that match the expected and original captions . This is accomplished by comparing n-grams of various lengths ranging from 1 to 4. As an example. "The man sat on a black chair," was the expected caption. "The man is sitting on the chair now ," is the original caption.

Bleu metric for 2-gram = Correctly expected Words (man, chair) / all expected Words (6 words) = 0.333.

C. TUNING EXPERIMENTS

Three experiments were conducted to choose the word embedding technique, the optimizer, in addition to the effect of the patch size on the proposed method.

EXPERMENT#1: CHOOSING WORD EMBEDDING TECHNIQUE

In this experiment, half of the Flicker Dataset samples were used to test three word

embedding techniques, i.e., GLOVE, WORD2VEC, and ELMO. VGG19 was used to extract image features, while the LSTM with sequence prediction is used to decode the images. The sample was divided into two parts: 80% a training and 20% for testing. The model was validated using the BLEU metric. The results are shown in Table 4.1.

Table 4.1. Word Embedding Experimnets.

Word embedding technique	**BLEU-1 gram**
GLOVE	0.42
Word2VEC	0.405
ELMO	0.418

The results show that the techniques have almost the same effect on the performance. Based on these results, GLOVE was selected as the embedding model of our system, and it will be used in the following experiments.

EXPERMENT#2: IDENTIFYING OPTIMIZER

After deciding on GLOVE as the embedding technique, this experiment was carried out to find the most effective optimizer. The same half of the Flicker Dataset samples that used in the previous experiment are used to test three optimizers: ADAM, SGD, and Nesterov. Also, VGG19 was used to extract image features, and LSTM with sequence prediction is used to decode the images. The sample was divided into two parts: a training subset and a testing subset. The model was validated using the BLEU metric. The results are shown in Table 4.2.

Table 4.2. Experiment with the Optimizer.

Optimizers	**BLEU-1 gram**
ADAM	0.49
SGD	0.42
Nesterov	0.45

As shown in Table 4.2, Adam optimizer outperforms the other two and it is the best option for our approach.

EXPERMENT#3: INVESTIGATING THE EFFECT OF BATCH SIZE ON THE PERFORMANCE OF THE DEVLOPED SYSTEM

To investigate the effect of the batch size, we selected random values for the batch size. The experiment was carried out using a sample of the Flickr dataset (50 % of the whole dataset). VGG19, with CNN at its core, is the image feature extraction method. The LSTM with sequence prediction image decoding method is used. The sample was split into an 80% training subset and a 20% testing subset. The model was validated using the BLEU metric, and ADAM was used as the optimizer. The results are shown in Table 4.3.

Table 4.3. Batch size Experimnet.

Batch sizes	**BLEU-1 gram**
30	0.44
16	0.45
50	0.54
80	0.56
100	0.48

The results show that instead of trying random values, other techniques should be used to tune the batch size; we used the Genetic Algorithm to do so. Following the preparatory experiments, the next section describes the experiments used to test the proposed method.

D. INVESTIGATION THE PERFORMANCE OF THE PROPOSED APPROACH

The performance of the proposed approach is investigated in this section. The default system is tested first, without any enhancements. The performance of the enhanced system is then demonstrated without batch size tuning. Finally, the enhanced system's performance is investigated and validated.

EXPERMENT#4: DEFAULT SYSTEM WITHOUT ENHANCING

In this experiment, we investigated the performance of the default system without our enhancements. The default system contained the default structure of CNN and LSTM with any additions.

The Flickr Dataset was used, and the image feature extraction method was VGG19. LSTM with sequence prediction is used to decode images. The sample was split into an 80% training subset and a 20% testing subset. The model was validated using the BLEU metric and obtained 0.42 score.

EXPERIMENT 5: ENHANCED SYSTEM WITHOUT BATCH SIZE TUNING

In this experiment, we investigated the performance of the developed system with some enhancements without batch size tuning. The used enhancements are:

- Adding Regularization term to convolutions.
- Adding dropout layer at fully connected layers.
- Default Batch: its value was 32.
- Optimizers: Adam.

The Flickr Dataset was utilized. VGG19 with CNN was the image feature extraction method. LSTM with sequence prediction was used to decode images. The sample was divided into two parts: a training subset and a testing subset. The model was validated using the BLEU metric. As shown in Table 4.5 (row: developed approach without batch size), the Bleu score was 0.51.

EXPERIMENT 6: ENHANCED SYSTEM VALIDATION

In this experiment, we investigated the performance of the developed system with all of the enhancements we discussed above to some studies mentioned in related works. All of the editions discussed in the CNN and LSTM structures were included in the enhanced system.

- Adding Regularization term to convolutions.
- Adding dropout layer at fully connected layers.
- Batch size: selected by genetic algorithm 64.
- Optimizers: Adam.

The Flickr Dataset was utilized. VGG19 with CNN was the image feature extraction method. LSTM with sequence prediction was used to decode images. The sample was divided into two parts: an 80% training subset and a 20% testing subset. The model was validated using the BLEU metric. As shown in Table 4.5 (row: developed approach without batch size), the Bleu score was 0. 73.

Table 4.4 shows some examples of the validation samples, where our model was able to describes the content of images very well.

Table 4.4. Examples of the validation samples, and the generated caption by the developed system.

Experiment	Image example 1	Image example 1
Experiment 1: Default system without enhancing	Two people are standing on mountain 	Man in red shirt is riding bike through the wood 
Experiment 2: Enhanced system without batch size tuning	Person is standing in front of mountain 	Man is rock climbing 
Experiment 3: Our developed approach	The mountain climber steep mountain 	Car drive down road marked by pine tree 

To sum up, the results of previous three experiments were compared the approaches of [4], [6], and [7], as shown in Table 4.5, it is obvious that the last version of the proposed system has significantly outperformed all other studied systems. In more detail, the Bleu scores for the existing approaches are not higher than 0.64, as shown in Table 4.5. After adding enhancements (adding regularization term to convolutions, adding dropout layer at fully connected layers, using Adam as optimizer) and tuning the batch size using genetic algorithm, the Bleu score in our model is 0.73.

Table 4.5. Comparing the performance of all version of the developed approach and the approaches of [4], [6], and [7].

Methods	**BLEU-1 gram**
ResNet [4]	0.62
InceptionNet [4]	0.63
EfficientNet [4]	0.64
TVPRNN [7]	0.59
CNN and GRU [6]	0.53
Experiment 4:The default VGG19	0.42
Experiment 5:Our developed approach without batch size	0.51
Experiment 6:Our developed approach	0. 73

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

In this research, our goal was to introduce an enhanced model capable of creating accurate captions for any input images. To achieve that, we have followed the following steps: First, conducted a review of related research to familiarize ourselves with the most recent research on the subject of image captioning and to gain the necessary knowledge. Second, a theoretical investigation of the tools required for image captioning related to language processing, image processing, and machine learning was conducted. Thirdly, we developed an efficient captioning model. Finally, The performance of the system was enhanced using some tuning techniques to formalize the deep learning such as Regularization, Dropout, Batch size, and Optimizer.

Various experiments were carried out to validate the proposed system. The results, according to the BLEU metric, showed that our model outperforms the some of the state of the arts models.

As future work, this system could be gradually developed to be used in real-time applications, such as videos. Also, the generated captions could be converted to voice to be used for other purposes.

5. REFERENCES

1 Tan, Y.H. and Chan, C.S., 2019. Phrase-based image caption generator with hierarchical LSTM network. Neurocomputing, 333, pp.86-100.

2 Ding, S., Qu, S., Xi, Y., Sangaiah, A.K. and Wan, S., 2019. Image caption generation with high-level image features. Pattern Recognition Letters, 123, pp.89-95.

3 Liu, M., Li, L., Hu, H., Guan, W. and Tian, J., 2020. Image caption generation with dual attention mechanism. Information Processing & Management, 57(2), p.102178.

4 Gupta, S., Agnihotri, S., Birla, D., Jain, A., Vaiyapuri, T. and Lamba, P.S., 2021. Image caption generation and comprehensive comparison of image encoders. Fusion: Practice and Applications, 4(2), pp.42-2.

5 Khamparia, A., Pandey, B., Tiwari, S., Gupta, D., Khanna, A. and Rodrigues, J.J., 2020. An integrated hybrid CNN–RNN model for visual description and generation of captions. Circuits, Systems, and Signal Processing, 39(2), pp.776-788.

6 Parikh, H., Sawant, H., Parmar, B., Shah, R., Chapaneri, S. and Jayaswal, D., 2020, April. Encoder-decoder architecture for image caption generation. In 2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA) (pp. 174-179). IEEE.

7 Yang, L. and Hu, H., 2017. TVPRNN for image caption generation. Electronics Letters, 53(22), pp.1471-1473.

8 Gupta, A. and Mannem, P., 2012, November. From image annotation to image description. In International conference on neural information processing (pp. 196-204). Springer, Berlin, Heidelberg.

9 Tariq, A. and Foroosh, H., 2015. Feature-independent context estimation for automatic image annotation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1958-1965).

10 Asghar, M.Z., Khan, A., Zahra, S.R., Ahmad, S. and Kundi, F.M., 2019. Aspect-based opinion mining framework using heuristic patterns. Cluster Computing, 22(3), pp.7181-7199.

11 Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A.C. and Berg, T.L., 2011. Baby talk: Understanding and generating image descriptions proceedings

of the 24th cvpr. Citeseer. Google Scholar Google Scholar Digital Library Digital Library.

12 Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

13 Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. “Incorporating Copying Mechanism in Image Captioning for Learning Novel Objects,” IEEE Conference on Computer Vision and Pattern Recognition, 2017

14 Cavalieri, D.C., Palazuelos-Cagigas, S.E., Bastos-Filho, T.F. and Sarcinelli-Filho, M., 2016. Combination of language models for word prediction: An exponential approach. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 24(9), pp.1481-1494.

15 Krizhevsky, I. Sutskever, and G. E. Hinton, \Imagenet classification with deep convolutional neural networks.," in NIPS (P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1106{1114, 2012

16 Yamashita, R., Nishio, M., Do, R.K.G. *et al.,2018.* Convolutional neural networks: an overview and application in radiology. *Insights Imaging* **9,** 611–629. https://doi.org/10.1007/s13244-018-0639-9.

17 Li, Z., Yang, W., Peng S. and Liu F., 2020, arXiv:2004.02806.

18 Cheng, D., Gong, Y., Zhou, S., Wang, J. and Zheng, N., 2016. Person re-identification by multi-channel parts-based CNN with improved triplet loss function. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1335-1344).

19 Vo, T., Nguyen, T. and Le, C.T., 2018. Race recognition using deep convolutional neural networks. Symmetry, 10(11), p.564.

20 Yu, Y., Si, X., Hu, C. and Zhang, J., 2019. A review of recurrent neural networks: LSTM cells and network architectures. Neural computation, 31(7), pp.1235-1270.

21 Gurjar, S.P.S., Gupta, S. and Srivastava, R., 2017. Automatic Image Annotation Model Using LSTM Approach. Signal Image Process. An Int. J, 8(4), pp.25-37.

22 Zaytar, M.A. and El Amrani, C., 2016. Sequence to sequence weather forecasting with long short-term memory recurrent neural networks . International Journal of Computer Applications, 143(11), pp.7-11.

23 Micah Hodosh, Peter Young, and Ju-lia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. Journal of Ar-tifical Intellegence, 47:853–899, 2013.

24 Yousefpour, A., Ibrahim, R., Hamed, H.N.A. and Hajmohammadi, M.S., 2014. A comparative study on sentiment analysis. Advances in Environmental Biology, pp.53-69.

25 ALQARALEH, S., Turkish Sentiment Analysis System via Ensemble Learning. Avrupa Bilim ve Teknoloji Dergisi, pp.122-129.

26 Baktash, A.Q., Mohammed, S.L. and Jameel, H.F., 2021, June. Multi-Sign Language Glove based Hand Talking System. In IOP Conference Series: Materials Science and Engineering (Vol. 1105, No. 1, p. 012078). IOP Publishing.

27 Kaladevi, P. and Thyagarajah, K., 2019. Integrated CNN- and LSTM-DNN-based sentiment analysis over big social data for opinion mining. Behaviour & Information Technology.

28 Rong, X., 2016, word2vec Parameter Learning Explained. arXiv:1411.2738.

29 Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L, 2018, Deep contextualized word representations. arXiv:1802.05365.

30 Goodman, E.D., 2014, July. Introduction to genetic algorithms. In Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation (pp. 205-226).

31 Kramer, O., 2017. Genetic algorithms. In Genetic algorithm essentials (pp. 11-19). Springer, Cham.