REPUBLIC OF TURKEY
ISTANBUL GELISIM UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

Department of Electrical and Electronics Engineering

IMAGE CAPTION AND HASHTAGS GENERATION USING DEEP LEARNING APPROACH

Master Thesis

Yahya Qusay Mahdı AL-SAMMARRAIE

Supervisor
Asst. Prof. Dr. Sevcan KAHRAMAN

Istanbul - 2023

THESIS INTRODUCTION FORM

Name and Surname	**:** Yahya Qusay Mahdi Al-Sammarraie
Language of the Thesis	**:** English
Name of the Thesis	**:** Image caption and hashtags Generation Using Deep Learning Approach
Institute	**:** Istanbul Gelisim University Institute of Graduate Studies
Department	**:** Electrical and Electronics Engineering
Thesis Type	**:** Master of Science
Date of the Thesis	**:** 18/01/2023
Page Number	**:** 66
Thesis Supervisors	**:** Asst. Prof. Dr. SEVCAN KAHRAMAN
Index Terms	**:** social media, deep learning, machine learning, text (caption) and hashtag generator, posts, images, NLP, VGG19, LSTM.
Turkish Anstract	**:** Derin Öğrenme Yaklaşımı Kullanarak Görüntü Başlığı ve Hashtag Oluşturma
Distribution List	**:** 1. To the Institute of Graduate Studies of Istanbul Gelisim University 2. To the National Thesis Center of YÖK (Higher Education Council)

Yahya Qusay Mahdi AL-SAMMARRAIE

REPUBLIC OF TURKEY
ISTANBUL GELISIM UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

Department of Electrical and Electronics Engineering

IMAGE CAPTION AND HASHTAGS GENERATION USING DEEP LEARNING APPROACH

Master Thesis

Yahya Qusay Mahdı AL-SAMMARRAIE

Supervisor
Asst. Prof. Dr. Sevcan KAHRAMAN

Istanbul – 2023

DECLARATION

I hereby declare that in the preparation of this thesis, scientific ethical rules have been followed, the works of other persons have been referenced in accordance with the scientific norms if used, there is no falsification in the used data, any part of the thesis has not been submitted to this university or any other university as another thesis.

Yahya Qusay Mahdi AL-SAMMARRAIE

…/…/2023

TO ISTANBUL GELISIM UNIVERSITY
THE DIRECTORATE OF GRADUATE EDUCATION INSTITUTE

The thesis study of deep learning titled as Image caption and hashtags Generation Using Deep Learning Approach. has been accepted as MASTER THESIS in the department of Electrical and Electronics Engineering by out jury.

Director *Asst. Prof. Dr. Sevcan KAHRAMAN*

Member *Asst. Prof. Dr. Yusuf Gürcan ŞAHİN*

(Supervisor)

Member *Asst. Prof. Dr. Kenan BÜYÜKATAK*

APPROVAL

I approve that the signatures above signatures belong to the aforementioned faculty members.

Prof. Dr. İzzet GÜMÜŞ

Director of the Institute

SUMMARY

Social media refers to the means of interactions among people in which they create, share, and/or exchange information and ideas. People use social media to stay connected with friends and family, to make plans, and to keep track of each other's day-to-day activities. Another way that social media is used is to advertise products or services. This type of marketing has been around since the 1800s. In fact, there were 2.934 billion monthly active users on Facebook alone.

Social media platforms want to make the experience of their users interactive, so they need to include captions and hashtags with posts. Individual users, on the other hand, do not always find it easy or possible to come up with captions and hashtags. Social media platforms are in need of captions and hashtags to make posts more interesting. There is a struggle when it comes to finding those two key elements. Whoever creates those captions and hashtags needs to be creative and keep up with social media trends in order for their posts not to be overlooked or lost among other posts in their feed. In this research, we have developed a system capable of creating appropriate texts and hashtags for social media platforms such as Facebook, Instagram, and other platforms that depend on these two elements based on the images or posts that are used on these platforms.

In this study, we built a way for generating text and hashtags based on images used as postings on social media networks such as Facebook, Instagram, etc. Our method is divided into four major stages. The first step is to obtain high-quality data. The acquired data is preprocessed in the second step. The third step is creating a text (caption) and hashtag generation technique utilizing Conventional Neural Network (CNN) and deep learning (DL) techniques. In this study, we look at several NN architectures like as VGG-16, VGG-19, and LSTM. Then, to construct the hashtag, we integrate natural language processing (NLP) to our technique. The evaluation phase is the final stage of this work.

We carried out experiments multiple times to find the best model, thus we run several tests to see which one had the highest accuracy and matching percentage between the text and hashtags and the images chosen.

Key Words: social media, deep learning, deep neural network, text (caption) and hashtag generator, posts, images, NLP, VGG19, LSTM.

ÖZET

Sosyal medya, insanlar arasında bilgi ve fikir oluşturdukları, paylaştıkları ve/veya değiş tokuş ettikleri etkileşim araçlarını ifade etmektedir. İnsanlar sosyal medyayı arkadaşları ve aileleriyle bağlantıda kalmak, plan yapmak ve birbirlerinin günlük aktivitelerini takip etmek için kullanmaktadirlar. Sosyal medyanın kullanıldığı diğer bir yol, ürün veya hizmetlerin reklamını yapmaktır. Bu tür pazarlama 1800'lerden beri vardir. Aslında, yalnızca Facebook'ta aylık 2.934 milyar aktif kullanıcı bulunmaktadir.

Sosyal medya platformları, kullanıcılarının deneyimini etkileşimli hale getirmek, yazıların yanı sıra başlıklar ve hashtag eklemeleri gerektiğini de istemektedir. Ancak, bireysel kullanıcıların bir başlık ve hashtag bulması her zaman kolay veya mümkün değildir. Sosyal medya platformları, gönderileri daha ilginç hale getirmek için altyazılara ve hastag'lere ihtiyaç duymaktadir. Bu iki temel unsuru bulmak oldukçazoidur. Bu altyazıları ve hastag'leri oluşturanların, gönderilerinin gözden kaçmaması veya diğer gönderiler arasında kaybolmaması için yaratıcı olması ve sosyal medya trendlerini takip etmesi gerekiyor. Bu araştırmada Facebook, Instagram gibi sosyal medya platformları ve bu iki unsura bağlı diğer platformlar için bu platformlarda kullanılan görsel veya gönderilerden yola çıkarak uygun metinler ve hashtagler oluşturabilen bir sistem geliştirilmiştir.

Bu araştırmada, facebook ve instagram gibi sosyal medya platformlarında gönderi gibi kullanılan görsellere dayalı metinler ve hashtagler oluşturmak için bir teknik geliştirilmiştir. Tekniğimiz dört ana aşamadan oluşmaktadır. birinci. aşama kaliteli veri toplamaktır. ikinci. aşama, toplanan verilerin önceden işlenmesidir. üçüncn. aşama, Konvolüsyonel Sinir Ağları (CNN) ve derin öğrenme yaklaşımını kullanarak metinler (altyazılar) ve hashtag oluşturma modeli geliştirmektir. Bu çalışmada, VGG16 ve LSTM gibi farklı NN mimarileri kullanılmistir. Ardından hashtag oluşturmak için tekniğimize doğal dil işleme süreci NLP yöntemi eklenmiçtir son aşama olaçkda değerlendirme aşaması eklenmiştir.

En iyi modeli elde etmek için birçok yöntem deneemiş, Bu nedenle (metin ve hashtag'ler) kullanılan görüntüler arasında en yüksek doğruluk ve eşleşme yüzdesini

elde etmek için birçok test gerçekleştirilmiştir. Elde edilen deneysel sonuçlar tablo şeklinde sunulmuştur.

Anahtar kelimeler: sosyal medya, derin öğrenme, derin sinir ağı, metin (altyazı) ve hashtag oluşturucu, gönderiler, resimler, NLP, VGG19, LSTM.

ACKNOWLEDGMENTS

First of all, I thank **God** for His mercy on us and for helping me complete my master's degree successfully.

Second, I would like to thank my supervisor, **Dr. Sevcan kahraman**, for his assistance and helpful direction in completing my thesis. This thesis would not have existed without her.

My wonderful family worked really hard so I can continue my educational journey till I reached this priceless moment. So, I'd like to thank **my father**, **my mother** and **my brother** for their endless support and encouragement. I truly love and respect you.

I would like to thank **Dr. Ahmed Amin** with all my heart for his contribution and his assistance to me during my master's studies

Also, I would like to thank my friend **Ahmed H.Mohsen** for standing by my side and answering all my questions about the master's degree.

TABLE OF CONTENTS

CHAPTER ONE
BACKGROUND

CHAPTER TWO
CONVOLUTIONAL NEURAL NETWORKS (CNN)

CHAPTER THREE
RECURRENT NEURAL NETWORKS (RNN)

CHAPTER FOUR
RELATED WORKS

CHAPTER FIVE
THE PROPOSED TECHNIQUE

CHAPTER SIX
EXPERIMENT RESULTS

CONCLUSION AND FUTURE WORKS

ABBREDIVATIONS

ANN	**:**	Artificial Neural Network
DNN	**:**	Deep Neural Network
GPU	**:**	Graphics Processing Unit
GRU	**:**	Gated Recurrent Unit
LSTM	**:**	Long-term, Short-term Memory
ML	**:**	Machine Learning
NE	**:**	Name Entity
NLP	**:**	Natural Language Processing
NN	**:**	Neural Network
CNN	**:**	Conventional Neural Networks
RNN	**:**	Recurrent Neural Network
TF	**:**	TensorFlow

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Social media is considered a great way to get a message out to the public, as it allows for showcasing of talent and expertise without even leaving the house. For example, Instagram, with over 1 billion users worldwide, is one of the most popular social media networks. However, Instagram posts may not contain a caption or hashtags. A sentence used to describe an image is known as a caption, which can be used on Facebook, Instagram, Twitter, and other social media platforms. A hashtag, which is a word or phrase with no spaces and a sign (#) at the front, categorizes posts on social media. On Instagram, hashtags must be placed within the caption box or in the comment section below the photo, when scrolling through the feed in order to see them. This may be confusing for some users who use other social media platforms like Facebook or Twitter, where hashtags can be placed anywhere on the post, as opposed to being contained in captions or comments.

However, many images are posted with inappropriate captions that lead to trouble and attract negative attention. It is suggested that more creativity should be employed when posting; copying other people's captions should be avoided; personalized captions should be used; and sometimes, the hashtags used are not appropriate for the image that is to be published on the social media platform.

To address the difficulties of creating captions and hashtags, a model is proposed to be built that can generate captions and hashtags based on the image that is to be shared on social media platforms. Instagram is used as the main social media platform for the current technique. Conventional Neural Networks (CNN) trained on images with their caption datasets are used to generate accurate captions and hashtags based on the image used in the post by the technique.

- **Instagram**

Instagram is a social networking app where images and videos are shared with friends and followers. It was created in October 2010 by Kevin Systrom and Mike Krieger. In 2012, over 100 million active monthly users were reached by Instagram. As of January 2017, Instagram had 1 billion monthly active users.

There are various ways to use Instagram, such as uploading images and videos directly to a profile page, posting them to specific accounts, tagging others in posts, commenting on other users' posts, following other users, and receiving updates from their content.

Instagram posts typically contain two components: the image, text, and hashtags. Instagram's filters and editing tools can also be taken advantage of. These allow special effects to be added to photos and videos, and Instagram filters can be used to search for hashtags.

- **Machine Learning (ML)**

The idea of machine learning is to automatically analyze data through model training, which makes use of artificial intelligence and allows machines or systems to think on their own and even expand their understanding over time (Géron, 2017). Similar to how people learn, ML starts by detecting patterns and data, and then, depending on the case situation, instructions are followed before judgments are made to solve a particular program. Knowledge is generally represented in ML through decision trees, neural networks, support vector machines, examples, and sets of rules.

- **Deep Learning (DL)**

Deep learning is defined as an advancement of ML, and is based on artificial neural networks and has several hidden layers. The term "deep" in "DL" is used to indicate the network's layer number. DL is an artificial intelligence principle that assists in recreating or emulating the learning processes used by humans to acquire various sorts of information (Ian Goodfellow, Yoshua Bengio, 2017). Simply defined, it is a method for machines to execute tasks using predictive analysis. The method used to generate text includes CNN and can be applied to different applications such as face recognition, image classification, etc. In our technique, CNN is described and how to use it.

A type of artificial intelligence algorithm that mimics how neurons work together in our brains is a neural network. Many layers of nodes connected by weighted edges constitute a neural network. Each node represents a neuron in the brain, and each edge represents the connection between two neurons..

- **Conventional Neural Network (CNN)**

A conventional neural network (CNN) is a more advanced sort of neural network that functions in far more than two layers. Advanced mathematical models are employed in order to handle complicated data. In simpler terms, CNN aid in tasks based on how well the human brain processes, such as image identification and processing that is especially intended to process pixel data.

The Neural Network with Convolutions (CNN) works by taking an image, providing itself with some weights depending on the image's several objects, and then identifying them against each other. This study provides further information regarding DL..

- **Natural Language Processing (NLP)**

Natural language processing is a branch of artificial intelligence that improves computers' ability to interpret and process human languages (Goldberg, 2017). This is significant for enhancing computers' understanding of human-level language. ML methods are used in NLP, and recent advances in ML, such as DL, have deeply changed how computers deal with issues such as language translation, text summarization, text production, and image caption generation.

- **Image caption generator**

The technique of creating written descriptions for images is known as "image captioning." The goal of image captioning is to automatically generate descriptions for a given image. Captions are generated using both NLP and computer vision (Pranoy Radhakrishnan 2017). With the improvements in deep neural network models in the past few years, automatic image captioning has become an attractive research field. Many factors make image captioning important, such as intelligent computer-human interaction, image search engine development, and automatic image captioning can all use this technique. Image Caption Generation is based on the functionalities of CNN and RNN. After the CNN algorithm, image captioning and NLP are used. The primary function of CNN is to extract objects and other spatial patterns from the input image, but RNN is useful with any sort of sequential data supplied to it. RNN and CNN are frequently referred to as the encoder and decoder, respectively. Along with CNN, NLP is utilized to produce image captions. LSTM are specialized RNN that allow information to be preserved.

- **Research Methodology**

The short methodology for the proposed technique consists of four stages: data collection, training CNN, evaluation, and the use of the trained model to generate captions and hashtags. In data collection, the data is collected from the Kaggle website as images and text files. Then, after data is collected, the dataset is trained by employing CNN on the gathered dataset to construct a model that can create descriptions and hashtags for the photos provided. The

training CNN is analyzed using following factors in the evaluation step: loss error value and accuracy.

- **Research Problem**

The caption and the hashtags are considered one of the most important factors for the success of the post in social media and helps to attract the attention of the person to those post. Therefore, people usually try to create several captions and hashtags until they reach the successful content for the post, and they sometimes hire content writers for this, which takes a lot of time, effort, and money. The technique that will be created provides text or captions and hashtags for the posts that are created on social media platforms.

- **Research Objectives**

  - **Main Objective**

The primary goal of the research is to develop a system that generates captions and hashtags for posts that need to be utilized on social media platforms such as Instagram.

  - **Specific Objectives**

The specific objectives of the project are:

- find some data that we can use to build the technique.
- Develop image captioning and hashtag generation techniques.
- Generate a caption and hashtag based on the used image.

- **Research Objectives**

Saving time, effort, and money by giving a good image description and hashtag to non-expert caption and hashtag writers.

- **Scope and Limitations of The Work**

- A DL algorithm is used.
- The proposed technique focuses on the content of the image.
- The caption and hashtag technique works with English images.

- **Equipment and Tools.**
  - good internet connection.
  - laptop with a very fast and high-speed processor.
  - Dataset collection (kaggle)
  - Tools and programming language
  - Python editor: vs.code
  - Google colab.
  - TensorFlow: TensorFlow is a Python-based deep ML framework.
  - Keras library.

- **Thesis Organization**

The rest of the thesis is organized as follows: The basis of the work is discussed in the second chapter, which includes ML, neural networks, DL, and NLP. Chapter 3 discusses CNN, Chapter 4 discusses RNN, and Chapter 5 reviews the related work of the research paper, including the image caption generator and hashtag generator. The methodology for the proposed technique is described in Chapter 6, as well as the dataset, the suggested technique (neural network settings), experimental design, and evaluation measures. Chapter 7 shows and examines the experimental results; it also covers evaluator methods and displays evaluation results, as well as samples of the created captions and hashtags. Finally, Chapter 8 contains the conclusion and future work

CHAPTER ONE
BACKGROUND

This chapter introduces ML and examines ML methods and applications. The following chapter gives background information about CNN. Then, DL is defined by clarifying the definition and methods of DL, as well as deep CNN. The following section describes RNN, very well known as the "RNN." RNN has a number of NN designs, for example, long-term short-term memory which is known as (LSTM). The concept of recommendation systems is then described and applied to tackle different problems. Finally, the image caption and hashtag generation system is discussed and how current RNN may be used to produce images or post content based on the problem it needs to solve.

1.1. Machine Learning

Machine learning has evolved over several decades, from past techniques to present techniques (Géron, 2017). Earlier, ML techniques began with finding patterns and evolved towards other computer concepts and ideas without the need for humans to advise them on how to carry out their given tasks. Artificial intelligence specialists intended to expose computers to fresh data by giving them repeating tasks, and the machines adapted over time. The robots enhanced their dependability by learning from previous experiences or doing tasks they had previously completed. As a result, scientists were motivated to learn even new scientific findings.

1.1.1. Applications of machine learning in social media

There are many applications of ML concepts in which these applications use artificial intelligence to perform requests, depending on the area of application. Social media has become an important part of our everyday lives, and it is also widely used in marketing and communication. Companies are trying hard to make social media more accessible and helpful while avoiding negative effects. Artificial intelligence (AI) is supporting them in this attempt by developing better apps and algorithms.

ML is a type of artificial intelligence that can learn without being programmed. It enables machines to analyze data and make decisions automatically. This may be used to improve the user experience in social media applications. It could, for example, automatically delete inappropriate information or spam for users.

Security is another useful application of AI in social media. ML assists social media companies in identifying and removing fake accounts. Fake accounts are commonly used for illegal reasons, such as promoting businesses that sell dangerous items or scams that take full advantage of emotionally or financially weak people.

Just like spam content, fake news is also popular. To detect fake news, social media platforms use ML tools and algorithms that use pre-recorded data to assist machines in detecting similar content. Fake news is much more dangerous and unethical than spam. Fake news may be instantly eliminated from any communication channel with the aid of appropriate reporting methods.

The last of all applications of ML in social media revolves around paid promotions. Paid promotions are the core idea behind ML in social media. Social media networks promote sponsored promotions more than organic marketing. They are paid advertisements. Such platforms can identify the appropriate target population and create such advertising thanks to ML.

1.1.2. Important components of machine learning

ML has three major components that must be present in every transaction. A representation, which is a method of selecting how knowledge is represented, is one of the components. The other aspect is how knowledge is evaluated in order to determine the outcome of a process. The final major component is optimization, which refers to how programs or solutions are developed throughout the search process.

1.1.3. Types of Machine Learning Systems

ML systems are grouped into broad categories depending on whether they are trained under human supervision (supervised, unsupervised, semisupervised, and reinforcement learning) (Shalev-Shwartz & Ben-David, 2013). These criteria are not always exclusive; they may be combined in any way that is seen as fit. Unsupervised ML involves comparing new data points to previously known data or implying patterns from training data.

Earlier, as explained, there are four types of ML, and each of them will be briefly explained as follows:

1.1.3.1. Supervised learning

Supervised ML algorithms are used wherever it is necessary to learn from new data and use it to make predictions about the future. The process begins with an analysis of previously studied data, and then learning algorithms are used to predict the values of the program's output. After a suitable amount of training using the known data has been completed, supervised learning algorithms help target new inputs. A traditional supervised learning problem, such as the classification of spam filters, is a wonderful example, as it learns to classify incoming emails after being trained with multiple samples of emails and their classification (hams and spams). Another common example is predicting the price of an automobile based on a collection of features (mileage, age, brand, etc.), which is referred to as "regression." To develop the system, dozens of instances of datasets, including their predictors and labels must be supplied.

1.1.3.2. Unsupervised learning

An alternative approach is Unsupervised learning. Unsupervised learning approaches are also employed, in which the computer works with no prior information. Even though the outcome is unknown, hidden data structures that can support prediction are discovered by the system. In this way, the method can help detect undiscovered buried data structures. If a large amount of data about a blog's visitors is known, a clustering method should be used to see if groups of similar people can be uncovered. The algorithms are barely told which category a visitor belongs to; relationships are discovered on their own. For example, it may be discovered that 40% of the guests are boys who enjoy comic books, 20% are sci-fi fans, and so on. If a multilevel clustering algorithm is employed, each group may be split into smaller groups.

1.1.3.3. Semi supervised learning

The output of information is generated using both known and unknown input by semi-supervised ML methods, which helps in enhancing learning accuracy. Typically, a little amount of supervised data and a greater amount of unstructured data is employed by this method. For example, photo-hosting sites such as Google Photos are considered a semi-supervised learning environment, so when all of your family photographs are uploaded to Google Photos, it immediately detects that the same person A appears in images 1, 5, and 11, while another person B shows up in photos 2, 5, and 7. This is the unsupervised part. Now all the system needs is that you inform the system who these persons are. It just needs one label per person and can name everyone in every photo, which is great for finding photos.

1.1.3.4. Reinforcement Learning

The fourth way is reinforcement ML, which is a completely different beast. Reinforcement learning is done through interactions with the program's environment by performing actions and detecting faults through trial-and-error search tactics, and with delayed incentives to evaluate the features required to enhance learning (Shalev-Shwartz & Ben-David, 2013). The learning system, referred to as an agent in this context, may track the environment, choose and implement actions, and get rewards in exchange. It must then learn on its own what is the greatest technique, known as a policy, to maximize reward over time. A policy specifies what action the agent should take in a particular scenario.

1.2. Conventıonal Neural Network (CNN)

Convolutional neural networks, often known as CNNs or ConvNets, are a type of neural network that uses convolution in its processing. Convolution serves to isolate and extract certain properties, as a result, CNNs have several applications in both computer vision and image processing. Features extraction increases pattern recognition in a variety of applications, from facial recognition in smartphones to navigation modules in self-driving cars. Applications are encouraged in identifying the area of concern and remaining components by advanced learning and appropriate training data sets. There are lots of techniques to quicken the procedure and hence produce better results. When utilizing such techniques, the hidden layer components are given to the CNN with additional adjustable possibilities, which can increase error checking.

1.3. Deep Learning (DL)

"Deep learning," a subtype of ML, helps in creating and designing data processing according to how a human brain should work. The relevant data from a scenario is first identified and then prepared for analysis (Ian Goodfellow and Yoshua Bengio, 2017). Analytical models are then created based on the chosen algorithm. The model is then trained upon the supplied datasets, with rounds repeated until the machine understands the patterns and underlying concepts. The final stage of the procedure is the training of the model to provide the desired output while also discovering new potential areas of use. This procedure is critical to understanding how DL works. Techniques that approximate how a real brain might attempt to turn the information into highly abstract and collected representations are used to process the input data. Each layer individually processes the raw images as well as the videos stream that is also provided as input, and then effectively encoded to complete the learning process.

As a result, the system may be modified to predict or perform more complicated actions on larger and newer data sets. Applications of DL include image processing, video stream processing, object classification, and the improvement of objects in older photographs. Handwriting identification and analysis, Alternate method usage include several modern programs with recommender system and content analysis components. ML methods can be "supervised" or "unsupervised," which indicates that the learning process is supervised if indeed the output categories are known; otherwise, it is not. Supervised ML is based on statistical learning theory.

DL is a mathematical framework for learning representations from data. These layered representations are (usually always) developed using neural network models in DL. The basic principles in DL are influenced by our understanding of the brain. There is no evidence that the brain employs learning processes equivalent to those implemented within the modern deep-learning models.

1.4. Recurrent Neural Network (RNN)

The production process of data sequences is a common application for neural networks. This is accomplished via recurrent neural network, also known as RNN. RNNs employ sequential data, which is an improved variant of CNN in which each layer's data is expected to be independently. RNN operates across all input items using the same set of operations and functions, hence the outcome is entirely dependent on earlier inputs. This is a sort of neural network with storage that can support incredibly lengthy sequences.

1.5. Long-Term, Short-Term Memory (LSTM)

Long Short Term Memory (LSTM) layers are recurrent units that are commonly used in deep neural networks. LSTM is a complicated unit, also referred as a cell, that is designed as a replacement for the basic neurons that were previously used in RNNs. LSTM may be seen as a layer style that can be blended with other layer kinds, such as a dense layer. It is an engineered solution that is not said to be biologically inspired. In current RNNs, the LSTM cell is often used. Technically, LSTM employs various sorts of transfer functions. The first kind of transfer function is the sigmoid. This sort of transfer function is used to build gates inside the unit. Another kind of transfer function is the hyperbolic tangent (tanh) function, which allows researchers to modify the LSTM outcome.

The RNN topology is similar to that of the LSTM network, but the LSTM network has more repetition modules and requires more operations. (Aggarwal, 2018) The advantage of the LSTM network is its ability to maintain long-term connections. Activities that aid in the LSTM network's ability to memorize more include:

- Forget gate operation: This is a task that occurs when the input is gathered from the current time and reproduced out from previous time step in the type of a combined value.
- Update gate operation: This demands concatenating data from several timestamps stated in the preceding step. The concatenated values are passed via a tanh function, which creates prospective values by feeding them through with a sigmoid function, the results of which are picked from possibilities.

This is managed by analyzing cell state and using tanh functions to seek for actions at a few unique operations. To calculate the output values of the next step, the values from the current time step and the previous timestamp are processed through a sigmoid function.

1.6. Natural Language Processing (NLP)

Natural language processing (NLP) is a branch of artificial intelligence that deals with understanding human language. It utilizes programming techniques to develop models that can recognize language and categorize information. The use of deep learning (DL) and other machine learning (ML) methods has greatly improved the ability of computers to handle tasks such as language translation, text summarization, and semantic analysis. This makes it easier for people to work with algorithms and large amounts of data. The collection and storage of this information on computers aids in the advancement of NLP. NLP enables computers to easily read, understand, and interpret text and speech. Unlike humans, computers communicate through binary operations of zeros and ones, making NLP crucial for resolving language ambiguities and incorporating numerical structures that are useful in a variety of data applications.

There are various techniques that illustrate the functioning of NLP. Context extraction is one of the key techniques used in natural language processing. Other techniques include content categorization, document summarization, content alerts, duplicate detection, search, and indexing. Speech-to-text and text-to-speech interactions allow computers to understand human language by converting voice commands into text, and text commands into speech.

Another application of NLP is machine translation, which facilitates speech translation from one language to another. All these applications aim to convert natural language inputs into computer commands by processing language using algorithms and programming languages.

Text Generation

Text generation is a technique of developing digital methods that allow computers to function as writers or speakers by using conditional language models. NLP is closely related to the frameworks that have a wide range of real-world applications. Text generation heavily relies on language models, which are a component of NLP techniques that involve probability distribution over a set of words. It can also be used to compute the probability of the next word in a sequence. Conditional language processing can assist in identifying a word sequence by generalizing the concept of assigning probabilities to a sequence, which also helps in determining the conditional context. Examples of RNNs used in text generation include long-short-term memory networks and gated recurrent unit sys

CHAPTER TWO

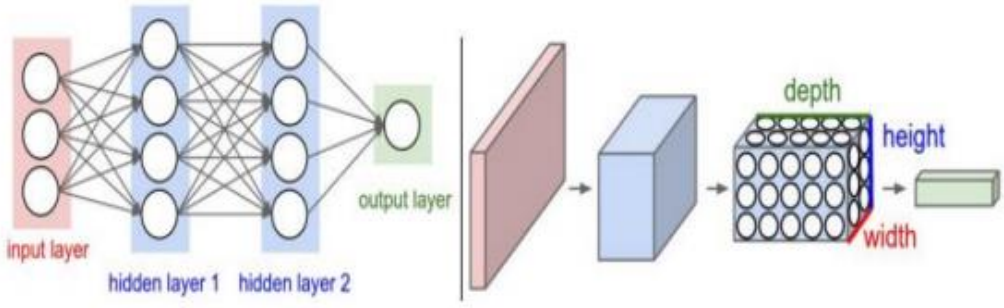
CONVOLUTIONAL NEURAL NETWORKS (CNN)

It is assumed that the network inputs of the Convolution Neural Network are structured samples. Applications involving computer vision, such as images and videos classification and edge detection. High accuracy has been demonstrated by CNN till now, making it useful for images recognition. A wide range of applications, including medical images processing, mobile encryption, and recommender systems, utilize images recognition. In CNN, convolutions refers to a specific functionality of convolutions, which is a method of linear procedure that mixes two main functions to produce a third functionality that displays how the structure among one function is modified by the other one.

2.1. Hyperparameter used in CNNs

The fundamental hyper-parameters are required to be known for the development and full comprehension of the functioning of CNNs. Three factors, height, width, and depth, have the CNN's convolution layer. The hyper-parameter indicated here has been used to establish the 3D dimensionality of this convolution network, which is essentially the numbers of neurons in the convolution layers.

.



(a) (b)

Figure 1. Neural network [1]. (a) Fully connected neural network, (b) convolutional neural network.

2.1.1. Receptive Field

Standard convolution is used by CNN filters to explore the whole image. The quantity of weights that must be solved for is considerably decreased because the filter size is actually significantly less compared to the image. The spatial scope of the filters is determined by the receptive field size. This approach is inspired by natural sciences, such as the receptive field in the minds of creatures such as cats and chimps, where it has been demonstrated that close image areas were linked with similar or close neurons in their minds. The diagram shows how the local receptive field of an input image is assigned to a number of neurons. After learning about the depth, it will be evident why it is being mapped to a bunch of neurons. It is crucial to remember that perhaps the receptive fields cannot be picky about just the depth of its own input and must work on the entire depth of the input. Local selection is only possible along the width and height coordinates of the input.

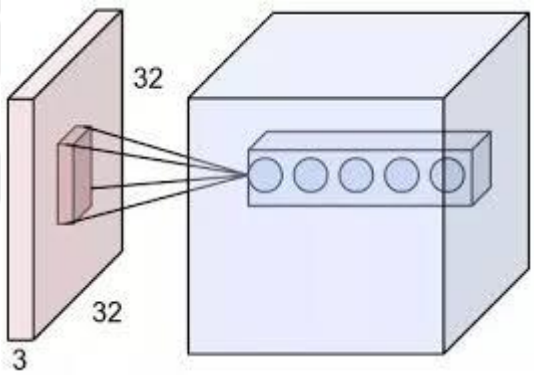


Figure 2. Receptive field idea illustration for a 32x32x3 image [2]

2.1.2. Depth

The number of individual neurons that process equivalent receptive fields of different weights is determined by the depth parameter of the convolutional layer. For example, a 5x5x5 filter might be used in traditional grayscale image processing. If the image is an RGB color image, the filter would be 5x5x3. The same receptive field is processed by many neurons, aiming to select and capture different information for the same input region. An individual output level is produced by each filter used in the input image (regardless of depth). As the network moves from input to output, the amount of filters increases as well as the depth of convolutional layers as the network moves from collecting basic features to more complex information within images. The number of hidden layers in a CNN should never be confused with the convolutional layer depth.

2.1.3. Stride

The depth of a filter is defined by the number of input planes and the step value through and down the images when the convolution is done is defined by the stride. The filter width, height, depth, and stride are used to build the 3D convolutional layer. A unit stride indicates that additional depth columns must be added for spatial areas of the image that are a unit distance away. Low stride values result in a greater number of resolutions per filtered image, with a large overlap in the receptive fields resulting in increased redundancy in weights. On the other hand, larger stride values provide lower resolution filtered images at the cost of an increased danger of quick loss of important information due to a large number of input factors contributing to a relatively limited set of parameters.

2.1.4. Zero padding

Zeros of the specified size are included as padding on both sides of the boundaries by zero padding. A zero padding of 1 for a two-dimensional locational image includes padding a row on top and bottom and a column on the left and right, raising the image's height and width by 2. Padding with such a value larger than zero is useful for preventing information on the image's edges from disappearing due to numerous convolutions. It also keeps the spatial dimensions of the convolutional layer output, sometimes known as the output volume. To know and appreciate the importance of keeping the spatial dimensions of the output, one needs to be aware with the pooling layers of CNNs.

2.2. CNN Architecture

Four different types of layers are used to construct CNNs. The convolutional layer (CONV) is the most important component, followed by the Rectified Linear Unit layers (RELU), pooling layers (POOL), and fully connected layers.

2.2.1. Convolutional Layer

The first layer is utilized to extract the different features out from the input images. The mathematical activity of convolutional neworks between the input images and a MxM filter is carried out by this layer. By sliding the filter over the input images, the dot product between both the filter and the areas of an input image with respect to the filter's size is determined (MxM). The resulting "feature map" provides information about the image, such as its edge and corner. This features map is then passed on to subsequent layers, which learn various characteristics from the input picture. The output is transmitted to the next layer after applying

the convolution operations to the input by the convolution layer in CNN. The convolutional layer in CNN benefits greatly because they keep the pixels spatially connected. The neurons in the convolutional layers are organized in 3D, taking into consideration the data's height, breadth, and depth.

2.2.2. RELU Layers

Non-linearities must be introduced into the CNN CONV layers for the network to learn complicated non-linear surfaces. As a result, the non-linear activation function RELU must be directly included as a layer after each CONV layer. RELU activations were chosen over other activation functions, such as the logistic sigmoid, because they do not saturate and eliminate gradients at the end, are zero-centered, and do not suffer from the disappearing gradient issue.

2.2.3. Pooling Layers

A convolutional layer is usually followed by a pooling layer whose main goal is to reduce the size of the convolved feature map in order to reduce computational charges. This is accomplished by reducing the links between layers and operating independently on every feature map. There are several sorts of pooling procedures depending on the approach utilized. The features created by a convolution layer are essentially summarized by it. The biggest piece from the feature map is used in Max Pooling. Average Pooling computes the average of the components in a specified image section size. Sum pooling computes the total sum of the components in the designated section. The pooling layer is typically used to connect the convolutional layer with the FC layer. This CNN model generalizes the features obtained by the convolution layer and supports networks in identifying the features independently. Calculations in a network also are decreased as a result of this.

2.2.4. Fully Connected layers

The Fully Connected (FC) layer, which includes weights and biases as well as neurons, is used to integrate neurons from different layers. These layers are often placed prior to the output layer and represent the final few levels of a CNN architecture. All input nodes are linked to and support all output nodes in fully connected (FC) layers, which are hidden layers. As a result, a fully connected layer may be thought of as a specific example of a convolutional layer that produces an output volume of 1 x 1 x K, where K is the entire number of neurons in the FC layer, and the receptive field of the filters is similar to the spatial dimensions of the input. The two are related, which makes it easier to construct the FC and CONV layers for CNNs in

an identical way. The structure of a typical CNN will now be evaluated once all the levels included in a CNN design have been covered. FIG displays a common CNN architecture.

.

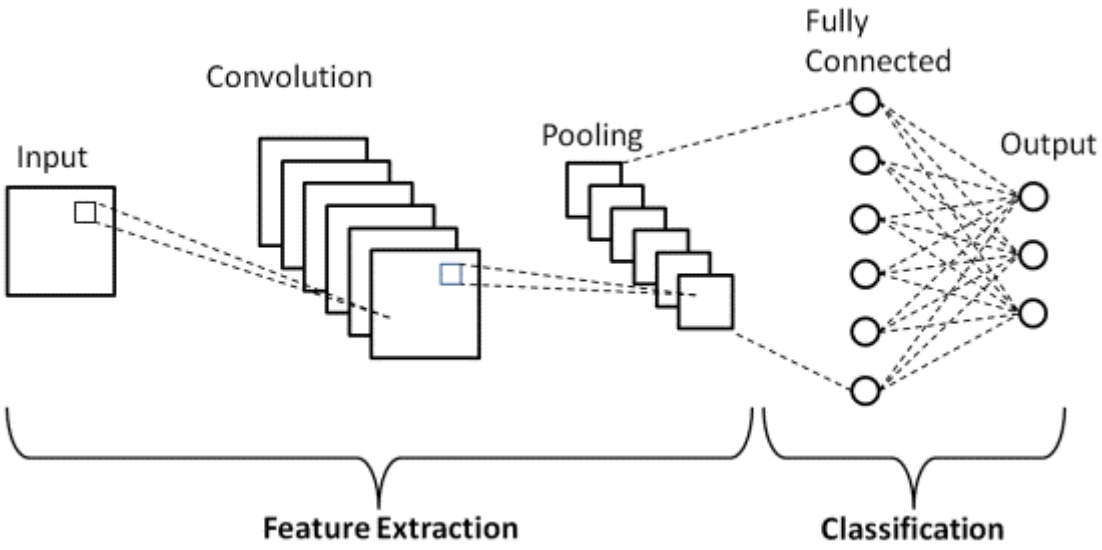


Figure 3. A common image classification issue requires a CNN architecture. [3]

The POOL layer is often not applied after the CONV and RELU levels due to the fact that many convolutions, each with a smaller receptive field, are typically chosen over using a single CONV layer, each with a more wide receptive field. With the added advantage of having fewer parameters overall, utilizing various CONV layers with extremely small receptive fields has the same result as applying a single convolutional filter with such a big receptive field. Smaller filters use a lot less parameters to achieve the same task. Also, additional CONV layers with smaller filters to perform the same task would deepen the CNN architecture and provide more nonlinearity into the input, improving classification outcomes. Regarding these benefits, if the spatial coordinates of the input to the CNN are extremely high and the output volume has to be decreased, a CONV layer with a large receptive field can be utilized in the first layer.

CHAPTER THREE

RECURRENT NEURAL NETWORKS (RNN)

Recurrent Neural Networks (RNNs) are ANNs in which neurons can create cyclical connections among themselves as well as interact with other neurons in the same layer. RNNs are a type of neural network that is both powerful and stable, and they are among the most attractive algorithms that are being used since they're the only ones that contain internal memory. RNNs, like so many other DL approaches, have a long and rich history. They were invented in the 1980s, yet their true potential has only just begun to be understood. RNNs have been pushed to their limits by rising computing power, massive amounts of data to operate with, as well as the 1990s finding of LSTM.

RNNs can store important details about the input they gained due to their internal memory, allowing them to estimate what will happen next with great accuracy. As a result, they are a collection of ML algorithms for series data, voice, textual, financial details, music, videos, climate, and other sequential data. When compared to other algorithms, RNNs can learn a lot more about a series and its surroundings.

In the other words, RNNs are indeed a sort of neural network that may greatly aid in the modeling of sequence data. RNNs, which are developed from feedforward networks, behave similarly to human brains. Simply expressed, RNNs can predict outcomes from sequential input that other algorithms cannot.

RNNs can be used when the spatial content of each individual frame is less significant and there is a succession of data and the temporal dynamics that connect the data.

To fully comprehend RNNs, we must first grasp normal feed forward neural network and sequential data. Sequence data is simply data that is organized in such a way that related elements follow each another. Two examples include financial statements data and the DNA data. Time-series data, which is just a sequence of data points given in numerical order, is perhaps the most prevalent type of sequential data. The cyclical interconnections are critical for catching the sequence of prediction of output, in which the present output is reliant not just on the input sequence but also on previous outputs..

A feed-forward neural network only transmits information in one way, from the input layer to the output layer via the hidden layers. Data travels immediately over the network. Feed-forward neural networks have little recall of the information they receive and perform poorly

in prediction because it simply evaluates the current input, a feed-forward network has no concept of temporal order. It simply cannot recall anything other than its teaching from the past. An RNN's information is spun in a loop. When it makes a decision, it considers both the current input and what has been learned from previous inputs. Two types of data are input into an RNN: the present and the recent past. This is vital because the data sequence contains important information about what will happen, which is why an RNN can accomplish things those other algorithms cannot. Before creating an output, a feed-forward neural network, like all other DL algorithms, assigns a weight matrix to its inputs. It is important to note that RNNs apply weights to both past and present input. Over time, a RNN will alter the weights for gradient descent and backpropagation. The illustration in the diagram below depicts the differences in information flow between an RNN and a feed-forward neural network.

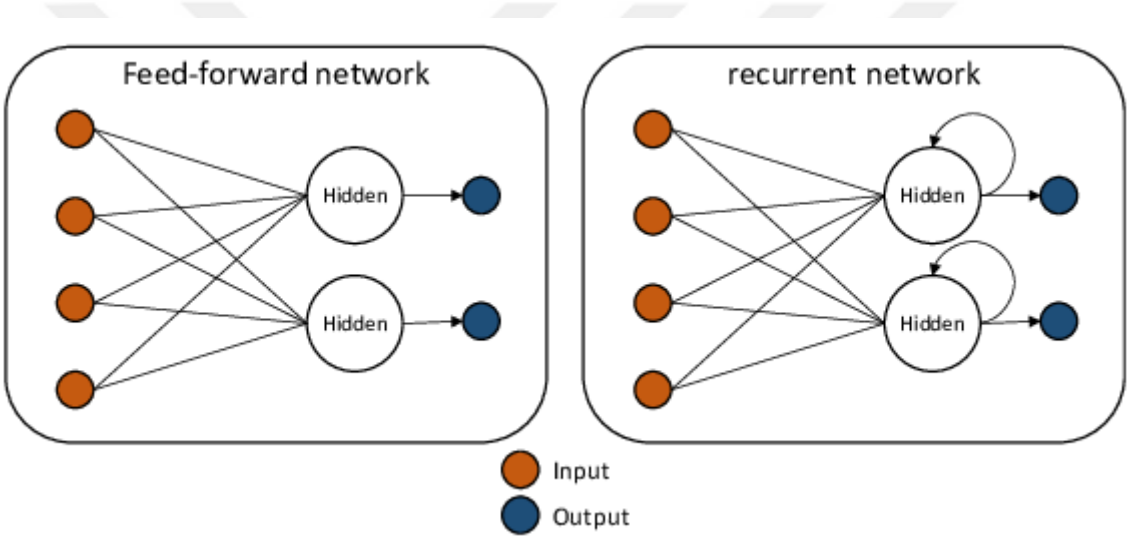


Figure 4. RNN with the characteristic cyclical connectivity. [4]

3.1. Types of RNN

The four most frequent forms of RNN are as follows:

- **One to One:** The far more basic RNNs have a single input and a single output. It works like a typical neural network, with fixed input and output sizes. The one-to-one application type has been included in image classification.
- **One-to-Many:** When given a single input, a one-to-many RNN produces several outputs. It accepts a set input size and returns a series of data outputs. It has uses in music generation and image captioning.

- **Many-to-one:** When a single output is needed from several input units or a sequence of them, many-to-one is utilized. A fixed output requires a sequence of inputs. Sentiment analysis is an example of a RNN of this sort.
- **Many-to-Many:** Many-to-Many is a method for generating a series of output data from a series of input units. This type of RNN is further subdivided into the two types: first, the equal unit size, which is the number of input and output units is the same in this situation. Name-Entity Recognition is a popular application, and second, the unequal unit, which is the size in this situation, the inputs and outputs have various numbers of units. Its use can be seen in machine translation

3.2. RNN and Backpropagation Through Time

Backpropagation (or backprop) is a fundamental ML algorithm. It is a technique for calculating the gradient of an error function when compared to the weights of a neural network. The method evaluates backwards through the gradient layers to calculate the fractional derivative of errors with regard to the weights. Backprop then utilizes these weights to reduce error margins during training. The output of any model is obtained through forward-propagation in neural networks and is determined if this output is accurate or incorrect in order to find the error. Backpropagation is just traversing any neural network backwards to determine the partial derivatives of the error with regard to the weights, this enables the elimination of this variable from the weights. Gradient descent, a technique that can continually shrink a given function, is used for these derivatives. The weights are then changed upward or downward to minimize the error. A neural network develops in this manner during the testing period. As a result, by employing backpropagation, the model's weights are changed during training. Backpropagation on an unrolled RNN is simply referred to as "BPTT." Unrolling is a network mapping and conceptualization tool that supports the understanding of what is happening. Backpropagation is typically handled automatically when establishing a RNN in commonly used software frameworks; however, the understanding of how it works is necessary in order to overcome problems that may arise throughout the construction process. A RNN may be considered to be a collection of neural networks which have been trained to follow one another via backpropagation. While unrolling all timesteps, BPTT propagates the error from the latest to the initial timestep. This enables the estimation of the error with each timestep and the adjustment of the weights. It should be noted that BPTT can be computationally costly when there are a large number of timesteps.

3.3. RNN and Long Short-Term Memory (LSTM)

Long-term memory enhancement is achieved through the use of LSTMs in RNNs, making them well-suited for learning from events separated by long time intervals. LSTMs are used to generate the layers of an RNN, and "weights" are added to the data, allowing RNNs to accept, ignore, or adjust the result based on the amount of weight given to the new information. LSTMs have been found to outperform RNNs in a range of NLP activities, including handwriting recognition, language translation, and image and video annotation. A typical LSTM cell has three gates: input, forget, and output. These LSTM cells, instead of using regular neurons, contain the hidden layers of an LSTM-based architecture. The input gate allows LSTM to preserve or overwrite the input from the previous hidden layer and the current input node by adjusting the effect of current input and past time-step output on the current cell state. Although LSTMs may satisfy temporal dependencies for more than 1000 time steps, it is occasionally necessary to discard previous data to prevent the introduction of undesirable dependencies during learning. The influence of prior cell states on the present cell state can be lessened or completely eliminated by the forget gate. The flow of information from the current cell state to the current concealed state output is controlled by the output gate. In other words, the layers of an RNN are constructed using the units of an LSTM. Due to LSTMs, RNNs can remember inputs for a long time. This is because LSTMs store information in memory, similar to a computer's memory. The LSTM has the ability to read, write, and erase data from its memory. This memory may be thought of as a gated cell, where the cell selects if it wants to store or erase information (i.e., whether or not to open the gates) according to the value it attributes to the information. Weights are used to assign priorities, which are also understood by the algorithm. This essentially means that it learns over time what information is important and what is not over time. A LSTM cell has three gates: an input gate, a forget gate, and an output gate. These gates decide whether to allow new input (input gate), discard the information since it isn't relevant (forget gate), or let it affect the output just at the current timestep (output gate).

CHAPTER FOUR

RELATED WORKS

An image for a short period of time is enough to completely describe it for a human, who innately performs this task and can generate extremely detailed descriptions. However, producing similar results with a computer has proven to be a difficult task. The works relevant to research and limitations of their works are reviewed in this chapter. Image captioning and hashtag generation have been applied in various applications such as recommendations in editing software, use in virtual assistants, image indexing, social media, and a variety of other NLP applications. Traditional and modern methods can summarize existing captioning methods. Describing an image using AI techniques, which combines computer vision with natural-language processing, is a difficult challenge, made even more difficult when asked to describe an Instagram post. Despite significant effort put into describing images, there is still a large amount of work to be done with captioning social media postings. Computer vision and NLP have been used extensively in describing the contents of images.

Anand Singh A DL system that includes a CNN with an LSTM of RNN was suggested by the author. A dataset available on Kaggle comprising 35,000 files was capable of being obtained, but because of computing limits, only 5000 images and descriptions were utilized for training the technique. Despite the fact that the algorithm did not produce improved outcomes, the process for generating content for Instagram images was built. This is a complicated challenge on its own, as a strong connection between images and the content that describes the things inside the images must be developed, handling images and words at the same time. The issue statement was to produce Instagram content automatically.

P.Mathur employed Deep Reinforcement Learning that attempted to improve his program to make much less complicated in order to operate on limited hardware resources, however the system trades off several accuracy.

A CNN-LSTM structure is utilized by the author **Shivam Gaur** to create hashtags for images. Traditionally, the architecture has been used to create descriptions in the way of NLP, however, this study uses the method to construct hashtags rather than phrases. Characters level of RNN can be trained on something like a corpus of individual stories to construct story-like captions from the hashtags generated above. A hashtag is picked as starter content from the

hashtag created in the previous process, and additional words are produced in sequence by utilizing character sequences of such a seed content.

Di Lu A new challenge of producing useful picture descriptions from the image and the hashtag was proposed. To fill this gap, a simple yet effective solution was presented. A (CNN-LSTM) model was trained on a given image to build a normal caption.

Chunseong Park C An unique captioning mechanism called Context-Sequence-Memory-Network (CSMN) was presented. This approach updates itself using the last memory system in the following order: i) Memories are employed as a storage for various sorts of contextual information; ii) To overcome the problem and grasp long-term information, predefined phrases are inserted into the memories. iii) A CNN is employed to improve component knowledge. The algorithm was trained on 1.1 million images on Instagram from 6,300 accounts.

4.1. Modern Captioning Methods (Neural Network Methods)

A CNN-RNN architecture is employed in recent studies that focus on neural network-based methods. The machine translation progress has been the inspiration for these neural network-based techniques. CNNs are used for encoding high-level visual information, while RNNs are used for decoding the CNN representation into a word sequence. The RNN, built using LSTM, receives the image information through the first interactions and predicts the following words given the previous word. Pretraining of the CNN on the ImageNet dataset is a common practice. LSTM or GRU are often used in RNN language models. Attention mechanisms have been included in the standard CNN-RNN framework by several approaches based on the CNN-RNN architecture. A localized image representation with several scales, produced by the encoder, is attended to by the decoder, which produces words sequentially. Two types of attention, between the decoder and the encoder, have been reviewed by Xu et al. Multiple stages have also been carried out by Fang et al. by using multiple instances learning to train visual detection for keywords that often appear in captions, and maximum-entropy training to build a model to create phrases including these terms. The zero-shot problem, when there are no image-sentence paired samples for training, was addressed by Hendricks et al. by generating phrase classifiers and a language model individually on unpaired picture and text data before combining them in a deep caption model trained on an image-sentence dataset.

4.2. Other similar work

Captions for Instagram photos are generated using a Keras CNN-RNN framework by Sejal Dua; the goal is to create captions that fit a given style and use specified words and phrases. A CNN and an LSTM that belongs to RNN are included in the model to achieve this. A 16-layer Oxford Visual Geometry Group (VGG) model pre-trained on the ImageNet dataset was employed to understand the content of the photos. The last layer of the CNN is removed to collect the extracted characteristics predicted by the model and feed them into an RNN decoder that creates captions.

A technique for generating hashtags was created by another person. Alec Morgan uses AI to automatically add hashtags to images. Programs that extract deep features from a new image and then utilize cosine similarity to discover multiple related photos in the training dataset are created by him. Additional computations are then employed to determine which of the hashtags on the images in the collection are most likely to be suitable for the new image. Finally, recommendations are rated, with the most certain recommendations coming first and the least certain ones coming last. To build the deep features, a pretrained model called MobileNetV2 was employed. It was discovered that doing so was optional in order to obtain perfectly good hashtag suggestions

.

CHAPTER FIVE

THE PROPOSED TECHNIQUE

Many chapters discussing how to deal with textual data have now been explained. Previously, a CNN was looked at for analyzing visual data. In this chapter, the design of an image captioning network by combining a CNN with a RNN will be shown. In other words, given an image, the network will provide a textual description of the image. The network will then be shown how to add hashtags. Therefore, the approach for our proposed technique is described in this chapter. First, the chapter describes the data collection and preparation processes. Following that, the features of the collected datasets are explained and several data samples are displayed. Following that, our image caption and hashtag generation technique, including the utilized CNNs and their parameters, is provided. Finally, the assessment measures utilized to evaluate our model are discussed in the chapter.

5.1. Methodology

It may appear difficult at first to design such a model, but it is actually quite easy. An encoder, which is a CNN that generates a language-independent intermediate representation of the image, is begun with. This one is followed by a decoder, which is a RNN that turns the intermediate representation into text.

As illustrated in Figure 5, the approach performed in this research consisted of the following main procedures:

- **Literature Review:** In this stage, related studies in the field of image caption and hashtag generation are examined and reviewed. In this step, the work into the perspective of other works is examined.
- **Data Collection:** In this part, the topic of the data used in the technique of creating texts and hashtags for images and ways to obtain them will be discussed, and also, ways to obtain new data will be explained.
- **Data Preprocessing:** Data preprocessing techniques include Data Cleaning, Data Transformation, and Data Normalization.
- **Develop Image caption and hashtag generator technique:** In this stage, a model that generates image captions and hashtags based on photos provided by the user, is created by employing CNN and LSTM. To get a desirable good

outcome, several experiment parameters and text preprocessing techniques are investigated. The good outcome demonstrates the usability and relevance of the caption and hashtag developed by our method.

- **Use the trained model to generate caption and hashtags**
- **Evaluation:** This section discussed the method that is used to evaluate the results. It has two measures that are used to evaluate it, The two-way evaluation involves both a manual and an automated evaluation. The manual evaluation is represented by human experts, whereas the automatic evaluation is represented by the loss error.

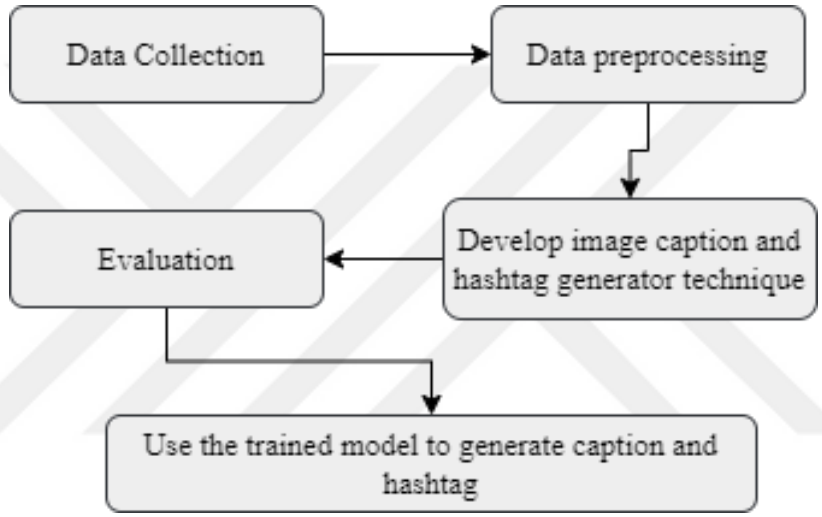


Figure 5. the methodology

5.2. Collecting and Preparing Datasets

When the search started, building own data was tried and we were thinking of getting images with their texts. Since a technique that helps in producing texts and hashtags for images is being built, Instagram was considered as the best social media platform to get data that helps in building the technique. So, images with their own texts were pulled from Instagram for many accounts using special software tools such as tools instagram-scraper.

Whereas, using the previous tool, a good number of data could be withdrawn and this data consisted of Instagram images and also a JSON file containing texts, hashtags, and other information related to each image pulled from a specific account. But several reasons that made us look for other ready-made data were faced and these reasons are as follows:

- The first reason is the long time that it will take to withdraw data for many accounts; one account takes from our time from half an hour to an hour to withdraw its data, and this time varies from one account to another depending on the number of posts in each account.
- The second reason is the large data size, because of the large number of Instagram accounts that we need to withdraw their data
- The third reason was because of the Instagram policy, where Instagram does not accept the withdrawal of different types of data, so that when we withdraw the data of 10 accounts, for example, Instagram suspends the account for a certain period of time, and it may take days to open it, or sometimes they close the account permanently. As 4 accounts were suspended for us for a different period of time, sometimes hours and sometimes days, during the process of withdrawing the data
- The fourth reason was in the long period of time that it would take to clean the data, and what is meant by cleaning the data is, for example, removing images without texts, removing texts without images, removing texts or images that are not in English, or removing texts that contain unknown symbols that may lead to problems with technical training, or removing junk information related to each image.
- The last reason is that after completing the data cleaning process for each account separately, the json files that we cleaned their data into must be collected into one file and converted to a csv file.

So, data that might be suitable for us was searched and found, which is the data that was taken from the Kaggle website (Kaggle, April 2010), (https://www.kaggle.com/). Users may search for and upload data sets via Kaggle, analyze and build models in an internet web browser data science environment, communicate with several other data scientists and ML professionals, and participate in solving data science challenges. Data called flicker8k from the Kaggle website was taken to use it to build the technique that is wanted. This data was chosen because:

- It is little in size. As a result, the model could well be quickly trained on desktop and laptop machines with minimum performance.

- The data has indeed been accurately labeled. For each image, there are five caption suggestions.
- The dataset is free and accessible.

Ways have been tried and found to build our own data, but because of the long time that it will take to build such data, and also the large size that this data will require, so the existing and ready-made data flicker 8k (Raman Shinde, 2019) was chosen. The entire process of creating a model requires the pre-processing and cleansing of data. Models that are more accurate can be created by understanding the data. After extracting the zip file from the download, the following folders and files will be discovered:

- The Flickr8k Dataset contains a total of 8092 JPEG images of various sizes and formats. 1000 are utilized for testing, 1000 for development (validation), and 6000 for training.
- Text or caption files outlining the train set and test set in Flickr8k text have been included. The Flickr8k.token.txt file has five descriptions or captions for each image, for something like a number of 40460 captions.

So, primarily two types of data are had. Captions and images (Text).

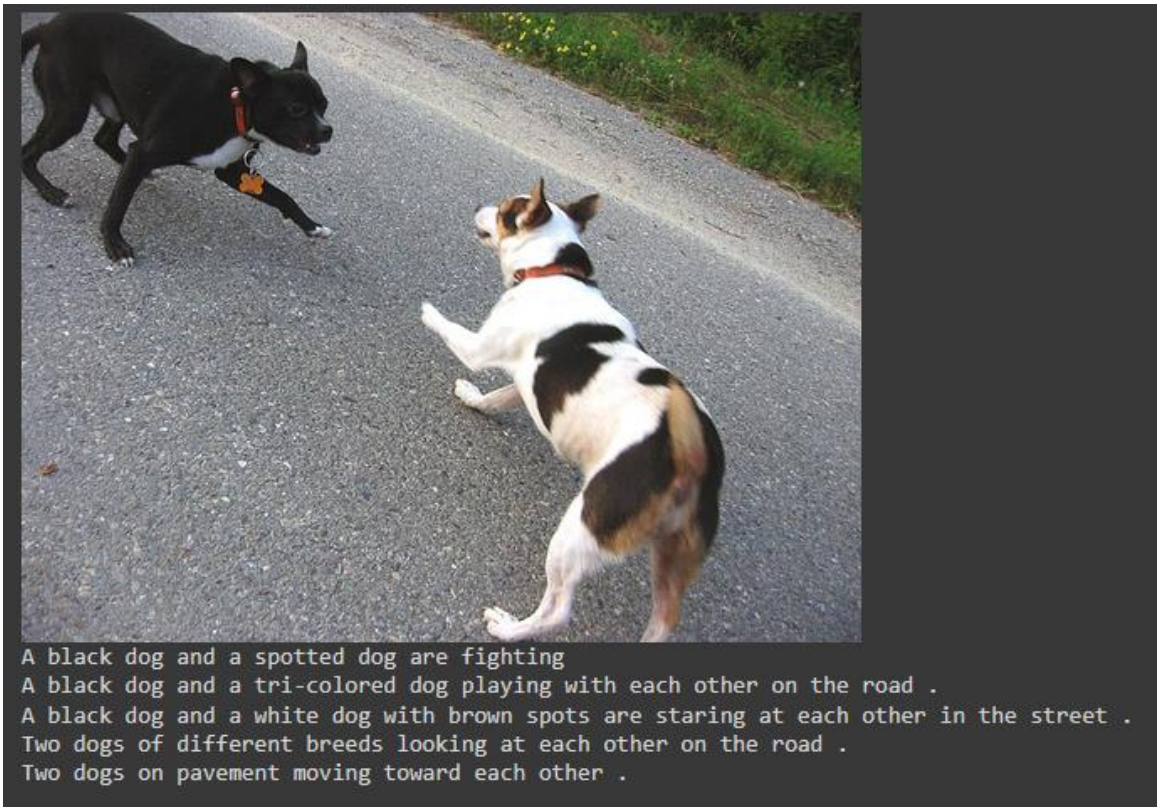


Figure 6. One example from the flicker 8k dataset

- **Another dataset**

A lot of data is available on the Internet that can be used to build our technique, such as the flicker 30k and coco datasets, and there is data on the Kaggle site, which are Instagram images with the texts of each image, that can be used to build the required technique.

Data set	data place (website, etc..)	Description	Dataset Size
Flicker 8k	Kaggle website	There are 8092 photos in all, with 5 captions with each image, for a total of 40460 texts.	1.19 GB
Flicker 30k	Kaggle website	Flickr30k Entities augments the 158k captions	9 GB
Instagram Images with Captions	Kaggle website	20k+ captioned images from celebrity Instagram accounts	4.14 GB

Table 1. Dataset information

5.3. Experiments Setup and Tools

5.3.1. Python 3.7.14

Python, version 3.7.14, is a computer programming language. This version is used in all our operations, including training, testing, and evaluation.

5.3.2. TensorFlow

TensorFlow (TF) is a Google Inc. open-source framework for data flow programming and ML (Tensorflow, 2019). It also includes a large mathematics library with too many mathematical formulas and equations. Neural networks are a type of ML technique, which are employed in our method. TF is applied in both research and production at Google and other industries.

5.3.3. Google Colab

Google Colab is essentially a web-based Jupyter Notebook, while Jupyter Notebook requires computer installation and can only communicate with local systems. resources, Colab is a powerful cloud platform for Python programming. Colab enables the writing of Python code in any web browsers, such as Google Chrome or Mozilla Firefox. These codes can also be executed in the browser without the need for runtimes or coding languages. Math formulas, charts, tables, illustrations, and other images can be used to develop any Python task notebook, making it stand out from the crowd. Python can also be used to generate visualization tools, and Colab will convert the script into a graphical result.

Free Colab members have accessibility to GPU as well as TPU runtimes for up to 12 hours. Their GPU runtimes consist of an Intel Xeon CPU running @ 2.20 GHz, 13 GB of RAM, a Tesla K80 accelerators, and 12 GB of GDDR5 VRAM. On the other hand, the TPU runtimes include an Intel Xeon CPU running at 2.30 GHz, 13 GB of RAM, as well as a cloud TPU capable of 180 teraflops of computing power.

5.4. Neural Network Parameters

The (CNN) and (RNN) kinds of deep neural networks (DNN) are employed in our technique. In our technique, the VGG16 is employed and then the VGG19 model from CNN is tried to see which is better, and the LSTM model from RNN is also employed. CNN is frequently used for image caption generation tasks due to its ability to accurately resolve image annotation issues. Two different models have been developed and tested for the feature extraction of image datasets. The input image size for both models is (224×224× 3), while the convolutional feature size for VGG is 4096. The two models have different abilities in extracting features from images.

- **VGG16:** is a model that has been pre-trained just on the ImageNet dataset using the Visual Geometry Group (VGG) OxfordNet 16-layer CNN. To categorize photos, the neural network VGG16 is implemented. The probability of each class that must be classified by the classification system is the output of VGG16. We used the VGG16's to extract the feature attributes for each image. For each image, we extract 4096 attributes.
- **VGG19:** To extract information from each image, we also employed a fully CNN based on the Visual Geometry Group (VGG) OxfordNet 19-layer.

The total number of weight layers in the VGG16 and VGG19 networks are 16 and 19, respectively.

However, several models are used in regards to the RNN parameters for the LSTM. Some of these parameters include the RNN size, number of layers, number of epochs, and optimizer.

A great amount of experiments and testing are performed before the final models and parameters are decided upon and utilized in the approach to produce captions and hashtags of high quality.

As described in Chapter 3, "CNN Applied to Image Classification," a CNN frequently ends with one or more fully connected layers which somehow summarize the feature maps from the previous convolutional layer into a 1D vector before the final softmax layer that classifies the image as containing a specific object. This 1D vector (the input to the softmax layer) in Visual the VGG19 of the Geometry Group has 4,096 elements, as seen in the Figure which illustrates a simplified representation of the VGG19 network. This vector can be understood as an image embedding, where the image is embedded in a 4,096-dimensional space. It can be imagined that two photos representing similar scenarios are embedded next to each other in vector space.

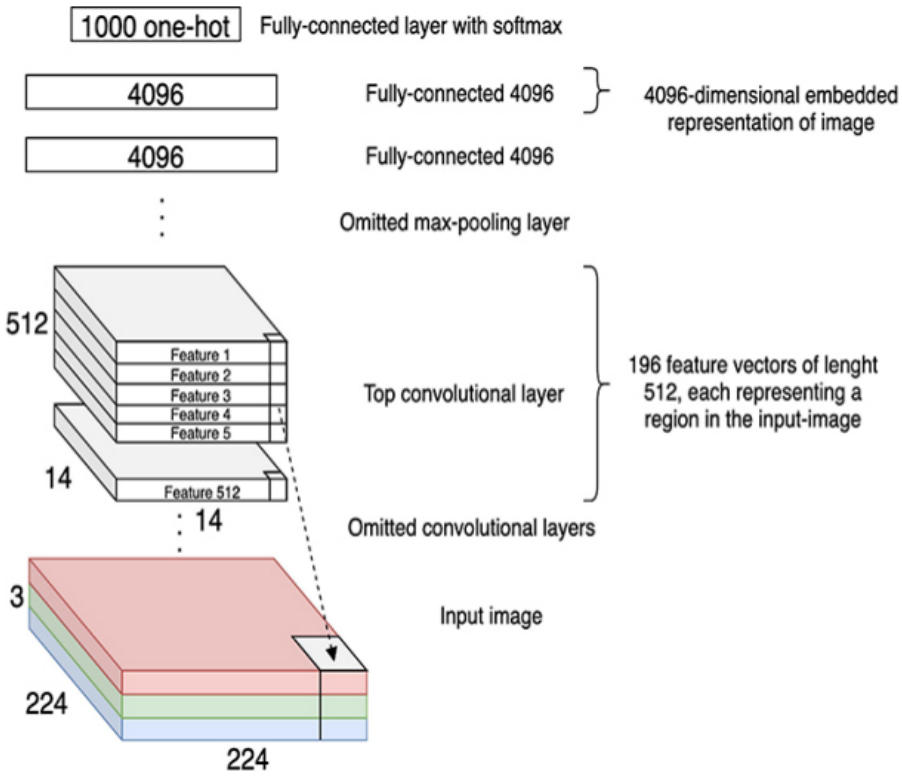


Figure 7. Simplified view of the VGG19 network where many layers have been omitted [5]

This vector can now be used as the context and fed directly into the RNN-based decoder network. It may also be used as the initial hidden state for the RNN-based decoder network. At first glance, it seems that an unnecessary constraint was imposed by requiring the number of units in the RNN (or LSTM) layer to match the size of the convolutional network layer. This would mean that the recurrent layer in VGG19 must have 4,096 units. However, this limitation can easily be overcome by adding another fully connected layer on top of the 4,096-unit layer. The fully connected layer at the top of the image captioning network has already compressed the many attributes into a single representation, so different sections of the 4,096-element vector do not correspond to different areas of the image. Each vector element contains information about every pixel in the supplied image. A more logical application of attention in the image captioning network would be to the top convolutional layer.

Top of Form

As it is known, the output of the convolutional layer in this type of network is a three-dimensional structure with two dimensions corresponding to the two dimensions in the image and the third dimension (the channels) representing feature maps for various types of features. This is also shown in Figure 16-3, Each of the 196 vectors corresponds to a distinct region in the input image, and the vector's 512 elements reflect the 512 possible types of characteristics that the network may have discovered in that region. When wanting to apply attention, using these 196 vectors as the context makes more sense because the attention mechanism can now attend to different sections of the input image by modifying the weights for the appropriate vectors.

5.5. Experimental Design

In this section, the construction of our technique will be explained. To construct a single neural network that generates Instagram captions and hashtags for images, our technique was built.

5.5.1. design the image caption

To produce captions for the photos, a CNN for image classification was mixed with a RNN for sequence modeling. The stages of creating this type of technique went through the following:

- the system started with entering the flicker 8K dataset that was downloaded from the Kaggle site, and then captions are loaded as values and images as keys in the

dictionary. And after that, a training, testing, and validation dataset files with headers as "image_id" and "captions" are created.

- The second stage is the loading of the 19-layer Oxford Visual Geometry Group (VGG) model, which had been pre-trained using ImageNet dataset. To gather the extracted features predicted by the model and feed them into an RNN decoder that creates captions, it was believed that doing this would enable the maintenance of important caption elements (hashtags), which would better match the captions that were trying to be produced. Then, after each caption is converted into an array of integers with a word for each index, each caption is encoded. To make training easier, a start sequence and stop sequence was included before each encoded caption.
- The third stage is the language model (RNN model) that has embedding and LSTM layers. In the image model, a dense layer is included. Then these two models are combined, and several parameters such as RNN size, batch size, number of layers, length, number of epochs, learning rate, optimizer, and other more stuff are used.
- These two models are compiled having an optimizer which is RMSprop and the metric accuracy.
- Specific values were trained on by the model to have higher accuracy and less losses, such as having the batch size equal to 512 and the epochs equal to 100. After the model was trained, a high accuracy of more than 0.7492 and a loss of less than 0.6183 was obtained.

Implementation	Model	The size
image_model	**Dense** **RepeatVector**	**512** **128**
language_model	**Embedding** **LSTM**	**128** **256**

Table 2: experiment parameters

5.5.2. design the hashtags

After completing the stage of creating captions related to the images, hashtags are started to be added to the images by us, and to get these hashtags, NLP is applied to the captions we obtained to extract the main keywords from the caption by us and add them as hashtags by us. It starts by deleting stop words, which are commonly used phrases (such as "the," "a," "an," or "in") that are meant to be ignored by search engines. These phrases aren't interested in by us because they are meant to eat up valuable database storage or consume processing time. They can easily be gotten rid of by keeping a list of phrases that users consider to really be stop words by us. Python's NLTK (Natural Language Toolkit) has a stopword collection in 16 languages. After removing the unnecessary words by us, the main keywords are left, which are highlighted or marked with the hashtag mark (#).

so, the figure below Figure 8 illustrates how this can be accomplished using an encoder-decoder design by us. Several articles (Karpathy and Li, 2014; Mao et al., 2014; Vinyals et al., 2014) independently presented similar designs around the same time as the sequence-to-sequence models for language translation were published.

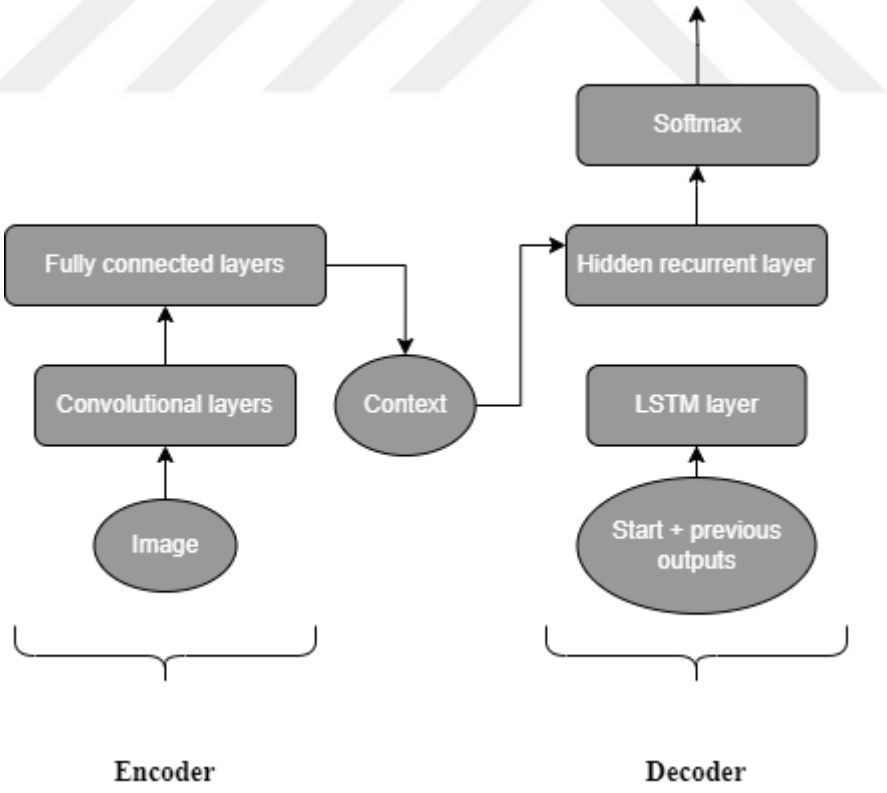


Figure 8. Architecture for image captioning network (encoder – decoder network for image caption)

In a more detailed way, a dataset, including photos tagged with written descriptions, is required for our application. The freely accessible flicker 8k dataset is made use of by us. Instead of training our network from the beginning to the finish by us, transfer learning for the convolutional layer is employed by us. This is done by employing a model that has been pretrained on the ImageNet dataset by us and implementing the VGG19 architecture. As previously explained, the fully connected layers are removed from the top of the network by us, and the output of the topmost convolutional layer is used to construct the context to which the attention mechanism will be applied by us. An optimization may be used since the weights for the VGG19 network do not need to be changed (it is presumed that the pretraining on ImageNet is appropriate) by us. Instead of passing the training image through the VGG19 network for each training example and each training epoch by us, each image may be run through the VGG19 network once and the vectors generated by the topmost convolutional layer saved to disk before training begins by us. That is, the encoder model is computationally simple during training since there is no need to process the image through all of the convolutional layers by us; instead, it just reads the feature vectors from disk by us. After the dataset that we want for the program is entered and the VGG 19 model is entered by us.

all of the photos can now be run through the network, feature vectors can be extracted, and stored to disk. To get the image file names, the dictionary is traversed. Every loop cycle performs processing on a single image and saves the feature vectors in a separate file by it. The image is preprocessed before being sent via the network. Because the image sizes in the Flickr 8K dataset are different and each image is unique, the file size must first be established by reading the file. The aspect ratio is calculated and the image is resized to a size where the shortest side is 256 pixels. The center 224x224 area of the generated image is then trimmed to get the input dimensions that the VGG-19 network wants by it. Finally, the VGG19 preparation function, which standardizes the data values in the image, is executed before the image is run through the network by it.

5.6. Evaluation Metrics

All evaluation metrics used in our work are described in this section. During the training phase, the training of the DL is analyzed by using metrics from our human expertise and loss error by us. During the review stage, the image caption and hashtag generator is analyzed by using metrics for readability and usefulness by us.

It is interesting to note that the best loss error value may not always produce the greatest results; instead, human review will be the final decider of the caption and hashtags' quality, readability, and usefulness by us.

5.6.1. Loss Error

During a training stage in ML and DL, predictions are made and compared to the truth. The loss function is an algorithm that compares predictions to reality by it. The loss error is computed for numeric output by it. The ML method attempts to decrease loss error during the training phase by adjusting NN weights.

The gradient descent optimization technique, which includes the optimizers RMSProp (Root Mean Square Propagation) and Adam, are employed in our work by us. The gradient descent is used by to find the optimal weights for machine learning in order to reduce the prediction error value during the training phase. Batch gradient descent, stochastic gradient descent, and mini-batch gradient descent are different versions of gradient descent, which differ by the method used to calculate the gradient using the given data..

The RMSProp and Adam optimizer are an extension of stochastic gradient descent. The RMSProp optimizer adjusts the learning rate based on the parameters by it. The learning rate for weight is divided by a running average of the magnitudes of gradients in weight by it. The Adam optimizer is an update to the RMSProp optimizer and it employs running averages of both gradients and gradient second moments by it.

5.6.2. Readability

Text readability refers to how well a text reveals its original meaning to a reader with few or no language mistakes, grammar, and phrasing by it. There are several elements that enhance readability:

- Font size and layout are physical variables (this was not considered in our evaluation).
- Less grammar mistake in text is better.
- Text Syntax: the less wrong words in the text, the greater the readability.
- Complex Word (Vocabulary Difficulty), the easier words in the text are better for reading.

Readability and relevance are evaluated using two factors: The first way is to read the generated text and estimate its readability by using a Python program (Textstat). The second way involves having human observers read and score the generated information for readability and relevancy by them.

When doing an evaluation, the Table displays the score and difficulty label for each created captions by it. The human observers choose "Easy" for produced caption if they consider the caption is easy to understand and may assign a score of 70 to 100, "Standard" if the rating is between 60 - 69, "Difficult" if the rating is between 30 - 59 by them.

5.6.3. Relevancy

The relevance of the created text to the image was determined by human participants, who were asked to choose "relevance" if it was relevant and "irrelevance" if it was not by them. The degree of match between the information obtained from the image and the reader's purpose, referred to as "textual relevance," is considered more interesting when it closely matches the viewer's desire and less relevant when it does not connect to the viewer's goal by them.

.

CHAPTER SIX

EXPERIMENT RESULTS

The results obtained after each training process carried out to establish the technique will be discussed, including the loss values and accuracy values obtained after the training operations. The evaluation methods previously discussed will also be discussed.

6.1. Results from trainning the model

The training of the model was shown in Table 3, including columns such as the training time for each epoch, the total number of epochs, the loss error, and the accuracy. The components are highlighted in the following points:

- Training time for a single epoch: An epoch is a whole sample repetition. The complete train-set is processed by the learning algorithm in ML. The number of seconds in this column reflect the typical training duration for one epoch.
- Training times: It is the overall training time in hours for each experiment alone.
- Total # of epochs This is the estimated number of epochs that are utilized in each experiment.
- Loss error the total loss value of the experiment; the lower value is desirable
- Accuracy It is better to increase the accuracy, as the more accuracy increases the accuracy of the texts we get

The model was trained with epochs of 5, 50, and 100, resulting in a reduction of the loss rate to 0.6183 and an increase in the accuracy rate to 0.7492 by it. The loss rate was reduced from 5.6897 to 4.9118 during the first training with 5 epochs and the accuracy rate was increased from 0.0758 to 0.1123. During the second training with 50 epochs, the loss rate was reduced to 1.8362 and the accuracy rate was increased to 0.4324 by it:

Training	Training time of one epoch	Training times	Total # of epochs	Loss error	Accuracy
1st Training	5,15 min	25.75 min	5 epochs	4.9118	0.1123
2nd training	5.03 min	4.1916 h	50 epochs	1.8362	0.4324
3rd training	4.62 min	7.7 h	100 epochs	0.6183	0.7492

Table 3: Results from trainning the model

6.2. Evaluators Ways

Human writers and Textstat evaluators (non-humans) are two of the types of evaluators who participate in the evaluation process. Human authors analyze both readability and relevance, as compared to Textstat, which just considers readability.

6.2.1. Textstat (Python Library) Evaluation

The readability of a created caption was assessed by using the Textstat package, a Python library, which calculates statistics that establish the corpus's readability, complexity, and grade level (Shivam Bansal, 2019). After being used Textstat, a human review was conducted to compare the results.

Textstat determines the readability of a given text by using the Flesch Reading Ease Formula (Farr, Jenkins, & Paterson, 1951). As displayed in Table 4, the algorithm returns the Flesch Reading Ease Score (FRES).

As stated in Table 5, if the score of the provided text is between 0 and 29, the difficulty is "very confusing", if the score is between 30 and 49, the difficulty is "difficult" and so on.

Score	Difficulty
90-100	Very easy to read and easily to understand
80-89	Easy to read
70-79	Fairly easy to read
60-69	Some times can be understood
50-59	Fairly difficult to read
30-49	Defficlt to read and understand
0-29	Very difficlt to read and understand (Very Confusing)

Table 4. The Flesch Reading Ease Score

Reading Ease Formula by Flesch

The Flesch Reading Ease Rating test, developed by Rudolf Flesch, is used to measure the readability of articles and evaluate how well readers understand English phrases. It calculates the lengths of words and phrases and includes weighting parameters. The results of the Flesch reading-ease rating test, shown in Table 4, are ranged from 0 to 100, with higher scores indicating material that is easier to read and lower numbers indicating material that is more difficult to read. The equation for the Flesch Reading Ease Rating test is shown in the figure below. It is used by the US government to measure the readability of articles.

$$206.835 - 1.015\left(\frac{total\ words}{total\ sentences}\right) - 84.6\left(\frac{total\ syllables}{total\ words}\right)$$

Figure 9. Eq: Flesch Reading Ease Formula (Flesch, 2015)

The ratios in the brackets are the formula's key components. All constants numbers were used to convert the rates into an understandable range.

Normalizing Difficulty Label

The ratings of the difficulties label by both human experts and Textstat were normalized and displayed in Table 5, as shown in Table 4:

Difficulty	Score	Filtered Difficulty
Very Simple, Simple, and Fairly Simple	70-100	Easy
Standard	60-69	Standard
Difficult and Extremely Rough	30-59	Difficult
Very confusing	0-29	Confusing

Table 5. Normalized Difficulty Label

6.2.2. Human Evaluation

In this experiment, a technique for creating content in the form of captions and hashtags for images on social media platforms such as Instagram, Facebook, etc. based on the image used by the user was developed. The Flicker 8K dataset was used to build the technique, and a CNN in the form of the VGG19 pre-trained model was applied to extract the properties and objects from the image. An RNN was used in the generation technique to produce content, employing various methodologies, architectural designs, and experimentation parameters, with LSTM being selected and applied. The size of the RNN, the optimizer, and the number of layers were experiment parameters. The appropriate structure was chosen by comparing the training loss and the accuracy of the produced content, resulting in an error rate of 0.6183 and an accuracy rate of 0.7492. The technique was specific to the English language and it was proposed to expand it to work with other languages such as Turkish and Arabic. After completing the programming of the technique, it was tested on a number of images. 74% of the people who were contacted were satisfied with the generated text and hashtags, while the rest complained about the weakness of the generated text, wanting it to be more detailed and include an explanation of the image.

6.3. Samples output of our technique

A number of images were tested on after the completion of the creation and programming of our technique, such as the following examples:

A dog runs through a a field .
#A #dog #runs #field

Figure 10. Test no.1

A black bird standing on a a the 's seeds .
#A #black #bird #standing #'s #seeds

Figure 11. Test no.2

CONCLUSION AND FUTURE WORKS

➢ Conclusion

A system that generates captions and hashtags for posts or images on social media sites such as Instagram, Facebook, etc. based on the image used by the user was developed in this experiment. The flicker 8K dataset was used to build the technique, and a CNN, specifically the vgg19 pre-trained model, was applied to extract the properties and objects from the image. An RNN was used to produce content, and various methodologies, architectural designs, and experimentation parameters were employed to achieve the desired outcome. The LSTM model was selected and applied to create this content, with the size of the RNN, the optimizer, and the number of layers being experiment parameters. Experiments were conducted to select the appropriate structure by comparing the training loss (measured errors) to the accuracy of the produced content, resulting in an error rate of 0.6183 and an accuracy rate of 0.7492.

A system that generated captions and hashtags for social media posts or images based on the image used by the user had been developed in this experiment. The Flicker 8K dataset was used to build the technique, and a pre-trained VGG19 CNN model was applied to extract properties and objects from the image. An LSTM RNN model was chosen and applied to generate content, with the size of the RNN, the optimizer, and the number of layers being experiment parameters. In the evaluation step, readability and relevance were measured by human experts and a Textstat approach for readability, with the results showing that 73% of the generated content was easy to read and 83% was relevant to the user's images or posts. As a summary, the technique worked best for generating content based on the used image, with the best results achieved by using the VGG19 and LSTM models with experiment parameters.

➢ Future Works

The following might be future works for our proposed technique

- The content (caption and hashtags) creation technique is specific to the English language, but it is proposed that it be expanded to operate with other languages such as Turkish and Arabic.
- The possibility of using larger data to increase the efficiency of the technique used

- Build a website that provides the content (caption and hashtags) generation technique to end user, so the user enters the image and then generate the required content.
- Improved our proposed method for creating other content element such as emojis and combining them with the caption.
- Developing our technique by building a different method for creating hashtags instead of our technique. Like our technique, it takes the important and main words from the created text and then converts these words into hashtags, but this technique can be developed, for example, by creating a dataset for images containing texts And hashtags for each image, and then make the machine train on that data to build new hashtags and texts for each image.
- Increasing the ability of our technique to produce multiple different captions and hashtags linked to image objects instead of producing a single content (caption and hashtags).

REFERENCES

Alakh Sethi (2020). Build your Own Object Detection Model using TensorFlow API. https://medium.com/analytics-vidhya/build-your-own-object-detection-model-using-tensorflow-api-f15ebd11b4e8

Anand Singh (2022). CNN-LSTM based Social Media Post Caption Generator, https://ieeexplore.ieee.org/document/9754189/

Andrej Karpathy (2015). The Unreasonable Effectiveness of Recurrent Neural Networks. http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Arslan Bajwa (2021). A Guide to Recurrent Neural Networks: Understanding RNN and LSTM Networks. https://builtin.com/data-science/recurrent-neural-networks-and-lstm

Aurélien Géron (2019) (Book). Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. https://www.amazon.it/dp/1492032646?linkCode=gs2&tag=oreilly2002-21

Bc. Jakub Kvita (2016). Image Captioning With Recurrent Neural Networks. https://core.ac.uk/download/pdf/44404652.pdf

Bin Shi • S. S. Iyengar (2020) (Book). Mathematical Theories of Machine Learning - Theory and Applications. Mathematical Theories of Machine Learning - Theory and Applications | SpringerLink

Cecelia Shao (2018). Implementing ResNet with MXNET Gluon and Comet.ml for image classification, https://medium.com/apache-mxnet/implementing-resnet-with-mxnet-gluon-and-comet-ml-for-image-classification-9bb4ad93a53f

Chen Yang (2021). Research in the Instagram Context: Approaches and Methods. https://www.researchgate.net/publication/349117428_Research_in_the_Instagram_Context_Approaches_and_Methods

Chu, Yan; Yue, Xiao; Yu, Lei; Sergei, Mikhailov; Wang, Zhengkui; Zhang, Yin (2020). Automatic Image Captioning Based on ResNet50 and LSTM with Soft Attention. Wireless Communications and Mobile Computing, 2020. https://www.hindawi.com/journals/wcmc/2020/8909458

Chunseong Park C, Kim B, Kim G. Attend to you: Personalized image captioning with context sequence memory networks. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 895-903).

Di Lu (2018). Entity-aware Image Caption Generation, https://blender.cs.illinois.edu/paper/imagecaption.pdf

Flicker 8k data set, https://www.kaggle.com/code/shweta2407/vgg16-and-lstm-image-caption-generator/data

François Chollet (2020). (Book). MEAP for Deep Learning with Python. https://www.amazon.com/Learning-Python-Second-Fran%C3%A7ois-Chollet/dp/1617296864

Gaur, Shivam (2019). [IEEE 2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW) - Macao, Macao (2019.4.8-2019.4.12)] 2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW) - Generation of a Short Narrative Caption for an Image Using the Suggested Hashtag. , (), 331–337. doi:10.1109/ICDEW.2019.00060

Guan, Zhibin; Liu, Kang; Ma, Yan; Qian, Xu; Ji, Tongkai (2018). Sequential Dual Attention: Coarse-to-Fine-Grained Hierarchical Generation for Image Captioning. https://www.mdpi.com/2073-8994/10/11/626/htm

Harshit Prasad (2018). GSoC 2018: RNN and LSTM Networks — Part II, https://medium.com/@harshit.prasad/gsoc-2018-rnn-and-lstm-networks-part-ii-96661bf24442

Jason Brownlee (2019). How to Develop a Deep Learning Photo Caption Generator from Scratch. https://machinelearningmastery.com/develop-a-deep-learning-caption-generation-model-in-python/

Jeff Heaton (2022)(Book). Applications of Deep Neural Networks with Keras,

Kapoor, Kawaljeet Kaur; Tamilmani, Kuttimani; Rana, Nripendra P.; Patil, Pushp; Dwivedi, Yogesh K.; Nerur, Sridhar (2017). Advances in Social Media Research: Past, Present and Future. Information Systems Frontiers, https://link.springer.com/article/10.1007/s10796-017-9810-y

Laurence Moroney (2020)(Book), AI and Machine Learning for Coders: A Programmer's Guide to Artificial Intelligence, https://www.amazon.com/Machine-Learning-Coders-Programmers-Intelligence/dp/1492078190

Lingxiang Wu (2019). Generating Descriptive and Accurate Image Captions with Neural Networks, https://opus.lib.uts.edu.au/bitstream/10453/140179/2/02whole.pdf

Magnus Ekman (Book). Learning Deep Learning: Theory and Practice of Neural Networks, Computer Vision, NLP, and Transformers using TensorFlow. https://www.amazon.com/Learning-Deep-Processing-Transformers-TensorFlow/dp/0137470355

Mahmut Yurt (2021). Deep Learning For Multi-Contrast Mri Synthesis. http://repository.bilkent.edu.tr/handle/11693/76446

Martin, E., Kaski, S., Zheng, F., Webb, G. I., Zhu, X., Muslea, I., … Kersting, K. (2011). Simple Recurrent Network. Encyclopedia of Machine Learning, https://link.springer.com/referenceworkentry/10.1007/978-0-387-30164-8_762

MK Gurucharan (2022). Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network. https://www.upgrad.com/blog/basic-cnn-architecture/

Mohammad Motiur Rahman (May 2021). Computerized classification of gastrointestinal polyps using stacking ensemble of convolutional neural network, https://www.researchgate.net/publication/351921164_Computerized_classification_of_gastrointestinal_polyps_using_stacking_ensemble_of_convolutional_neural_network

Motaz Alfarraj (August 2018). Petrophysical-property estimation from seismic data using recurrent neural networks, https://www.researchgate.net/publication/327612400_Petrophysical-property_estimation_from_seismic_data_using_recurrent_neural_networks

Oriol Vinyals Google (2015). Show and Tell: A Neural Image Caption Generator https://arxiv.org/pdf/1411.4555.pdf

P. Mathur, A. Gill, A. Yadav, A. Mishra and N. K. Bansode, "Camera2Caption: A real-time image caption generator," 2017 International Conference on Computational Intelligence in Data Science(ICCIDS), 2017, pp. 1-6, doi: 10.1109/ICCIDS.2017.8272660.

Ram Manohar Oruganti (2016). Image Description using Deep Neural Networks. https://scholarworks.rit.edu/cgi/viewcontent.cgi?article=10202&context=theses

Raman Shinde (2019). Image Captioning With Flickr8k Dataset & BLEU https://medium.com/@raman.shinde15/image-captioning-with-flickr8k-dataset-bleu-4bcba0b52926

Reza Bosagh Zadeh. TensorFlow for Deep Learning by Bharath Ramsundar, Reza Bosagh Zadeh. https://www.oreilly.com/library/view/tensorflow-for-deep/9781491980446/ch04.html

Seeliger, K., Fritsche, M., Güçlü, U., Schoenmakers, S., Schoffelen, J.-M., Bosch, S. E., & van Gerven, M. A. J. (2017). Convolutional neural network-based encoding and decoding of visual object recognition in space and time. NeuroImage. https://www.researchgate.net/publication/318454279_Convolutional_neural_network-based_encoding_and_decoding_of_visual_object_recognition_in_space_and_time

Sejal Dua (2019), Do it for the ‘gram: Instagram-style Caption Generator. https://towardsdatascience.com/do-it-for-the-gram-instagram-style-caption-generator-4e7044766e34

Shikha Gupta (2021). Step by Step Guide to Build Image Caption Generator using Deep Learning https://www.analyticsvidhya.com/blog/2021/12/step-by-step-guide-to-build-image-caption-generator-using-deep-learning/

Shuang Bai; Shan An (2018). A survey on automatic image caption generation. https://press.liacs.nl/students.mir/inspiration/A%20survey%20on%20automatic%20image%20caption%20generation.Neurocomputing2018.pdf

Sion Chakrabarti (2021). A comprehensive tutorial on Deep Learning – Part 1. https://www.analyticsvidhya.com/blog/2021/05/a-comprehensive-tutorial-on-deep-learning-part-1/

Stanford University, (Winter 2023). Hardware Accelerators for Machine Learning (CS 217), https://cs217.stanford.edu/weightlayers

Trushna Patel (2020). Object Detection Based Automatic Image Captioning using Deep Learning. https://bvmengineering.ac.in/NAAC/Criteria1/1.3/1.3.4/18CP809_Thesis.pdf

Van Hiep Phung (2019). A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets, https://www.mdpi.com/2076-3417/9/21/4500/htm#B26-applsci-09-04500

Yang, Min; Liu, Junhao; Shen, Ying; Zhao, Zhou; Chen, Xiaojun; Wu, Qingyao; Li, Chengming (2020). An Ensemble of Generation- and Retrieval-Based Image Captioning With Dual Generator Generative Adversarial Network. IEEE Transactions on Image Processing, https://ieeexplore.ieee.org/document/9226120

RESUME

Personal Information

Surname, name : AL_Sammarraie, Yahya Qusay Mahdi.

Nationality : Iraq

Education

Degree	Education Unit	Graduation Date
Master	**2 years**	**18/1/2023**
Bachelor		
High School		

Work Experience

Year	Place	Title

Foreing Language

Arabic

English

Publications

- “Image captions and hashtags generating using deep learning approach”, conference paper, IEEE research gate.
- “Designing fire detection and extinguishing systems”, b.sc degree thesis, Tikrit university, Tikrit-Iraq

Hobbies

Online games, traveling.