Automatic Generation Image Captions based on Deep Learning and Neural Network

1st Anjali Nara
department of computer Science
University of central Missouri
Lees's Summit, Missouri
axn50710@ucmo.edu

2nd Durga Sai Teja Thota
department of computer Science
University of central Missouri
Lees's Summit, Missouri
dxt10290@ucmo.edu

3rd Sowmya Nalini Koppolu
department of computer Science
University of central Missouri
Lees's Summit, Missouri
sxk74100@ucmo.edu

4th Rushitha Mudigonda
department of computer Science
University of central Missouri
Lees's Summit, Missouri
rxm58570@ucmo.edu

***Abstract*—We present an innovative approach to automatic image captioning using a Recurrent Long Short-Term Memory (R-LSTM) framework. This technique starts with an image and systematically constructs a detailed caption from a sequence of words. The core of this system is the R-LSTM architecture, which excels at capturing long-term relationships in data sequences.**

The procedure begins by preprocessing the image to isolate essential features, utilizing a Convolutional Neural Network (CNN) for this purpose. We then explore the R-LSTM setup in depth, highlighting how attention mechanisms are integrated to enhance the quality of the captions by considering the broader context of the image.

To evaluate our model, we perform tests using a recognized benchmark dataset and benchmark its performance against other modern techniques. Results demonstrate that our R-LSTM model achieves superior performance in generating high-quality captions compared to existing methods, according to established evaluation metrics.

We also address the significant challenges involved in developing an efficient automatic image captioning system. These challenges include the complexity of crafting captions that are both semantically rich and the high demand for extensive annotated datasets. Nevertheless, we argue that our approach represents a significant improvement in the field, potentially improving how visual content is accessed and utilized online.

In conclusion, our research supports the R-LSTM model as a valuable tool for enhancing automatic image captioning. We believe this work will encourage further research and development in this area, ultimately impacting the fields of image understanding, content accessibility, and digital media retrieval.

***Index Terms*—LSTM,CNN,NLP, Dataset,Word Sense Disambiguation ,VGG.**

[1]

I. Introduction

Numerous sources, including television, the internet, and news media, offer an abundance of visual content. While humans can easily understand these images without any textual context, machines require explicit descriptions to interpret them effectively.

Natural scene captioning is a key research area that enables the creation of image descriptions within the field of artificial intelligence [1]. This line of research is crucial for several applications, with major companies like Facebook and Google using it to analyze user behaviors, determine locations, and support related functions.

Understanding images requires discerning objects, their actions, and the interconnections among them. Machines often struggle with subtleties, such as recognizing that people are waiting for a train even if the train isn't visible in the image. It's vital for the generated descriptions to be not only grammatically correct but also semantically appropriate [2]. To fully grasp the details of a natural scene, feature extraction is necessary. This extraction generally divides into two main approaches: (1) Deep Learning-based methods and (2) Traditional Machine Learning-based methods.
Extensive labeled datasets like ImageNet, coupled with the power of deep learning, offer a significant advantage, particularly in the effectiveness of deep convolutional neural networks (CNN). The field of computer vision has witnessed substantial progress through image captioning, enabling computers to perform various tasks such as early education, video tracking, sentiment analysis, and aiding individuals with evident impairments. Ongoing research in artificial intelligence is increasingly focusing on advancing image captioning capabilities.

In this field, key goals include finding images, interpreting their connections, and extracting semantic details in natural language. Image captioning efforts frequently utilize template-based strategies, which necessitate outlining elements such as direct or indirect objects, their relationships, and their attributes.

The underlying structure for these methods is built around an

[1] https://github.com/naraanjali/Final_Project.git

encoder-decoder model, which involves two primary stages. First, image features are extracted through Convolutional Neural Networks (CNNs), which encode these features into structured embedding vectors. Then, the generation of descriptive text is typically handled by a Recurrent Neural Network (RNN) decoder. The widespread adoption of neural network-based approaches, which can generate new phrases, owes much to the effective representation capabilities of CNNs and the temporal modeling strengths of RNNs.

Recent developments in image description technology have shown significant promise for aiding visually impaired individuals in interpreting their surroundings, a testament to the growing field of computer vision [3]. Early techniques for crafting image captions combined statistical language models with static libraries of object classes. A notable innovation in this area involves an automatic geotagging system that uses a dependency model to extract location data from web pages associated with images. L and their team have enhanced this approach by introducing a network-scale n-gram method that compiles possible terms and combines them to form sentences, building coherent image descriptions from scratch. This language model utilizes the English Gigaword corpus, with hidden Markov model parameters refined based on these data. Descriptions are then produced by selecting the most likely nouns, verbs, contexts, and prepositions identified in the sentences. The goal is to categorize every possible area, applying a prepositional association function and using a Conditional Random Field (CRF) to predict image tags. Additionally, an object detection mechanism and 3D image analysis are employed to determine objects, characteristics, and connection points, which are then organized into semantic trees.

After learning syntax to create textual descriptions from semantic trees, the process was inverted, transforming these trees back into visual representations. Yagcioglu et al. developed a method that expands search queries to retrieve images from large datasets [4]. This technique combines the distribution of mentioned elements with the images fetched, representing one of the many indirect strategies devised to tackle the complexities of image captioning. After forming the expanded query, the method rearranges suggested descriptions by calculating the cosine similarity between the representation of each description and the expanded query vector. The final description for the image is then derived from the closest match in the dataset. Before the advent of big data and the widespread use of deep learning techniques, the efficiency and broad utility of neural networks had already been driving significant progress in the field of image description, opening up new possibilities.

II. Motivation

Image captioning is a dynamic field that lies at the crossroads of computer vision and natural language processing. Its primary aim is to equip computers with the ability to understand visual data and generate descriptions in natural language that are both accurate and meaningful, much like human communication. The process of creating image captions is complex and involves multiple capabilities, including object detection, understanding the scene, and modeling language. Initially, the system identifies key elements in the image, such as objects and individuals. It then uses this information to formulate a clear and concise sentence that accurately captures and describes the scene portrayed.

Image captioning serves multiple purposes, including aiding those with visual impairments, improving search engine results, and enhancing the user experience in various sectors. One practical use is the creation of product descriptions on e-commerce sites, which allows customers to search visually for items based on specific features. Additionally, image captioning is vital for the development of autonomous vehicles, robots, and other AI-driven systems that require a deep understanding of and interaction with their environments. By enabling robots to accurately articulate what they see, image captioning helps foster the creation of smarter, more adaptable systems capable of effectively navigating and interpreting complex settings.

Despite recent progress in the field of image captioning, several hurdles remain, including resolving ambiguities present in both language and visual data, handling scenes with numerous objects and actions, and incorporating contextual details into the captions. However, the potential advantages of image captioning are substantial [7], and continued research in this area is expected to lead to significant developments in artificial intelligence and its associated disciplines.

III. Main Contributions And Objectives

Non-functional requirements are critical aspects of a system that influence user experience without being directly linked to the specific functions the system performs. These requirements are measurable attributes such as response time, which measures the speed at which the system reacts to user inputs, or accuracy, which assesses the correctness of the system's calculations.

Key non-functional requirements for the system encompass:

- **Usability**: The system is designed to operate completely autonomously, removing the requirement for any manual user input.

- **Reliability**: The system benefits from enhanced reliability, a characteristic attributed to its development on the Python platform. Python is renowned for its stability and robustness, which contribute to the system's dependable performance.

- **Performance**: The system is being crafted using advanced high-level programming languages, along with cutting-edge technologies for both the front-end and back-end. This ensures that the client system benefits from a minimal response time.

- **Supportability**: The system is developed using high-level programming languages, complemented by sophisticated technologies for both the front-end and back-end. This configuration results in minimal response times for the client system.

IV. Proposed Framework

A. Proposed Framework

This paper presents a new method known as Reference-based Long Short-Term Memory (R-LSTM), which aims to improve the captions of query images by using reference data. The model, during its training phase, assigns varying weights depending on the association between the images and the corresponding words. It seeks to enhance the agreement scores between the generated captions and reference data from similar images, thus improving the accuracy of image recognition.

In the realm of natural scene captioning, Kiros et al. have developed an encoder-decoder architecture that merges image-text embedding models with multi-modal sentence generation frameworks. This setup functions similarly to language translation systems, producing descriptions for images on a word-by-word basis. The method involves encoding textual information with a specialized Recurrent Neural Network (RNN) known as Long Short-Term Memory (LSTM), while visual information is processed using a deep Convolutional Neural Network (CNN). The visual data undergoes optimization through a pairwise ranking loss and is then integrated into an embedding space augmented by LSTM hidden states that also encode textual data. In this space, a structured neural language model decodes the image features based on the associated word's feature vector, enabling the sequential generation of sentences.

Drawing inspiration from neural machine translation, Vinyals et al. utilized a deep CNN for image encoding and an LSTM within an RNN framework for decoding, thus aiding in the generation of descriptive captions from image features.

B. Benefits of the proposed system

Image captioning proves to be a valuable tool for individuals with visual impairments, providing them with the ability to grasp the content of visuals. Utilizing an AI-powered image caption generator, descriptions of images can be audibly conveyed to those with visual impairments, enhancing their understanding of the environment.

C. Life Cycle Model

In our project, the waterfall model was employed, as it encompasses five distinct steps:

- **Requirements**: At this phase, there is an understanding of the design, functionality, and goals, with all requirements being carefully documented.
- **Design**: During this stage, the specifications that arise from the initial requirements are analyzed, culminating in the development of the system design. This design, also known as software architecture, is essential in determining the necessary hardware, technology, and comprehensive system requirements.

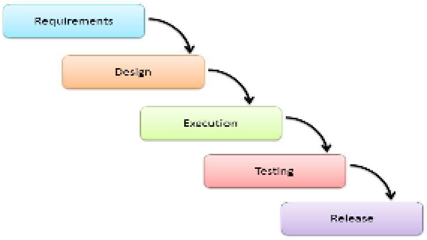

Fig. 1. Life cycle method

- **Execution**: Informed by the system design, the software is broken down into smaller, manageable units. This step initiates the coding phase, where the requirements previously defined are coded into these units and later integrated to create the complete software product.
- **Testing**: The product is thoroughly tested at various stages to verify that it is free from errors and meets all specified requirements. This testing aims to identify and resolve any issues that might arise when the client installs the software.
- **Release**: Following the completion of the testing phase and verification that the product is free of errors and fully compliant with requirements, it is deployed to the customer's environment or launched into the market.

D. DESIGN

1) Architecture: The leading method for creating image captions combines Convolutional Neural Networks (CNN) with Recurrent Neural Networks (RNN). This dual-input model processes images and their respective captions. Within the RNN, each layer processes a single word and the model learns to predict the next word by refining itself through the analysis of the caption data. Image features are derived using a pre-trained VGG16 model, saved to a file, and then matched with the captions. The extracted image features and outputs from the LSTM layers are then merged and fed into a decoding model to generate the final captions. The decoder's last layer is dimensioned to match the size of the vocabulary. To forecast the likelihood of each word, the model uses categorical cross-entropy, and the Adam optimizer is applied to update weights during the training phase.

E. Dataflow Diagram

Data flow diagrams are graphical representations that illustrate how data moves through an organization's information system. These diagrams show the processes involved in moving data from input to storage and eventually to report generation. There are two primary types of data flow diagrams: logical and physical. Logical data flow diagrams focus on the flow of data through the system to support business operations, detailing the what and why of data movement. Physical data

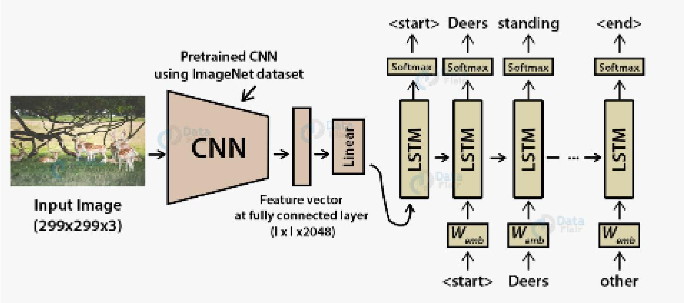

Fig. 2. Model-Image Caption Generator

flow diagrams, on the other hand, detail the actual implementation of this data flow, showing the specific hardware, software, and processes involved in the data movement.

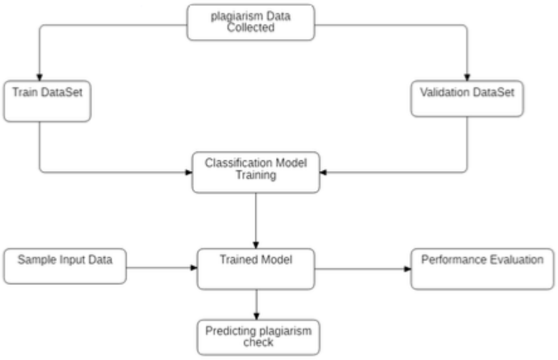

Fig. 3. Data Flow

Design engineering often incorporates the Unified Modeling Language (UML), a standardized notation used to draft software blueprints. UML is widely used as a language for

- Visualizing
- Specifying
- Constructing
- Documenting the artifacts of a software intensive system.

The Unified Modeling Language (UML) functions as a language that provides a set of vocabulary and rules for assembling words from that vocabulary to aid in effective communication. A modeling language is defined by its specific vocabulary and rules, which are designed to represent both the conceptual and physical aspects of a system. Through modeling, it is possible to achieve a thorough understanding of how a system operates and is structured.

V. IMPLEMENTATION

A. *Working of Project*

1) Dataset: Initially, our model underwent training on the Flickr30k dataset, comprising 31,783 images, each accompanied by five captions. However, challenges arose concerning the model's generalization, stemming from the limited number of training samples and the repetitive nature of the ”A man...” template in every caption. To overcome this, we transitioned to the more extensive MSCOCO (2014) training dataset [9], encompassing 82,780 images, each associated with five ground truth captions. For offline evaluation, we employed the Karpathy split3, a non-standardized yet widely utilized split in research, comprising 5,000 images.

2) Syntax Analysis: Syntax identification involves checking if a language follows its set grammatical structures. Techniques commonly used in this process include parsing, stemming, and lemmatization.

3) Semantic Analysis: In the field of Natural Language Processing (NLP), various algorithms are used to understand the intended meaning of text. Techniques such as Word Sense Disambiguation, Named Entity Recognition, and Natural Language Generation (NLG) are employed to facilitate this understanding.

B. *Convolutional Neural Network (CNN)*

Convolutional Neural Networks (CNNs) are widely utilized in visual recognition tasks, distinguished by multiple convolutional and fully connected layers. CNNs effectively leverage the two-dimensional structure of images through local receptive fields, shared weights, and pooling techniques, achieving translation-invariant features. Notably, CNNs are easier to train and have fewer parameters than other networks with comparable numbers of hidden layers. In this regard, the VGG network, a deep CNN designed for complex image recognition tasks, is available in versions with 16 or 19 layers. It shows consistent error rates across both validation and test datasets. The VGG network plays a pivotal role in extracting features from images that are crucial for caption generation. Additionally, Long Short-Term Memory (LSTM) networks, a type of recurrent neural network, are engineered to capture the nuances in sequential data, addressing issues such as vanishing or exploding gradients that traditional RNNs encounter. LSTMs feature a memory cell that retains information over extended periods, with gates that control the timing of updates to the cell state, allowing for adjustable interactions between the cell and its gates.

C. *Model deployment*

The model deployment phase consists of integrating the tested and validated model into a production environment, typically overseen by a data engineer or database administrator after the data scientist has verified the model's accuracy and defined its performance metrics. The deployment approach can vary based on the organization's infrastructure and the specific problems being addressed. Data engineers play a crucial role in establishing, testing, and maintaining the necessary infrastructure for efficient data handling, including adapting the model from high-level programming languages to lower-level languages that better fit the operational ecosystem.

For assessing the model's impact, data engineers often employ A/B testing to observe how users interact with the model, particularly in scenarios involving personalized recommendations, and to determine its alignment with business goals. In environments with smaller datasets, the deployment might be managed directly by the database administrator.

The deployment strategy might also hinge on whether the processes were initially conducted manually by a data science team with in-house IT resources or automated using Machine Learning as a Service (MLaaS) platforms. MLaaS platforms, such as Google Cloud AI, Amazon Machine Learning, and Microsoft Azure Machine Learning, provide a comprehensive array of services for data preprocessing, model training, testing, deployment, and prediction on cloud platforms, each offering different automation levels and capabilities.

Deploying through MLaaS often involves a high degree of automation, facilitating straightforward integration with specific company infrastructures or other cloud services through REST APIs. It is also important to consider the delivery mode of the analytics results, whether they are provided in real-time or at predetermined intervals, to enhance their effectiveness and relevance.

VI. GENERATION OF SENTENCE WITH LSTM

The method of generating sentences in neural networks employs the encoder-decoder architecture, which is typically utilized in machine translation tasks. In this setup, an encoder transforms sequences of words from natural language into distributed vector representations. These vectors are subsequently utilized by a decoder to generate a sequence of words in a target language. The primary objective during training is to improve the translation process to enhance the production of coherent sentences in the source language. When this approach is applied to image captioning, the emphasis transitions to improving both the length and clarity of the captions generated for particular images.

A. Strategy: matching the problem with the solution

In the initial phase of a machine learning project, it's essential to establish strategic objectives. This step includes identifying the problem, setting the scope of the project, and outlining a development strategy. A business analyst plays a key role in evaluating the feasibility of a software solution and identifying essential requirements. Simultaneously, a solution architect oversees the development process to ensure that these requirements are successfully integrated into the software. The main responsibility of the solution architect is to confirm that the requirements specified by the business analyst are accurately implemented and serve as the basis for the solution being developed.

B. Dataset preparation and preprocessing

Data serves as the foundation of any machine learning initiative. The second phase of these projects involves complex tasks including data acquisition, selection, preprocessing, and transformation, each consisting of specific procedures that must be meticulously executed.

C. Data Collection

The data analyst plays an integral role in the success of a machine learning project. They are tasked with identifying suitable data sources, gathering data efficiently, analyzing it, and using statistical methods to interpret the findings. The specific data needed varies according to the predictive task at hand, and determining the right amount of data required for a machine learning endeavor can be challenging due to the distinct characteristics of each issue. Selecting the right attributes for a predictive model is crucial as it hinges on their predictive power. Generally, acquiring as much data as possible is advantageous, as larger datasets tend to enhance the accuracy of model training. Additionally, augmenting internal data with publicly available datasets from platforms like Kaggle, Github, and AWS, which offer free datasets, can be beneficial.

D. Data visualization

Visual presentation of large data sets can greatly enhance comprehension and aid in analysis. It is crucial for data analysts to be proficient in creating diverse visual aids like slides, diagrams, charts, and templates. For example, a chart depicting several years of sales data can effectively highlight trends and patterns in sales performance.

E. Labeling

Supervised machine learning involves training a model on historical data that includes pre-defined answers, known as labels. This training requires the algorithm to be informed about the specific outcomes or attributes to detect within the data. Labeling data, especially in large datasets, is a labor-intensive task that demands significant effort to adequately train a machine learning model. For instance, if a model is designed to identify different types of bicycles in images, the dataset must be meticulously categorized and labeled. Engaging domain experts for data labeling can be an effective strategy to ensure the accuracy and relevance of the labels provided.

F. Transfer learning

Transfer learning offers a strategic alternative for managing large datasets by allowing the reuse of pre-labeled training data. This method involves utilizing the knowledge and insights gained from solving similar problems in different projects or by other data science teams. A data scientist assesses which parts of an existing training dataset can be adapted and applied to a new problem. This technique is especially prevalent in training neural networks for tasks such as image and speech recognition, image segmentation, and modeling human movement, where it helps accelerate the learning process and improve model performance without the need for extensive new data labeling.

G. Data selection

After collecting all necessary information, a data analyst strategically selects a subset of data relevant to the problem at hand. For example, if a company wants to understand the geographical distribution of its customers, it would not require personal details such as cell phone or bank card numbers. Instead, information like purchase history would be vital for building a predictive model. This chosen subset includes only those attributes deemed essential for the predictive model's development. In smaller data science teams, it is common for a data scientist to manage various tasks including data collection, selection, preprocessing, transformation, model construction, and evaluation.

H. Data preprocessing

Preprocessing is a crucial step in preparing data for machine learning. The goal is to transform raw data into a structured and clean format that is optimal for use in a machine learning model. This transformation helps data scientists achieve more accurate results from their models. The preprocessing phase involves various techniques such as data formatting, cleaning, and sampling, which are essential for refining the data.

I. Data formatting

Data formatting becomes increasingly important when dealing with data collected from diverse sources and individuals. Data scientists begin by standardizing the format of records, ensuring that variables are consistently represented for each attribute. This standardization also applies to attributes expressed through numeric ranges. Maintaining data consistency is crucial for improving the accuracy and reliability of machine learning models.

J. Data cleaning

During this phase, data scientists implement several procedures to improve data quality by removing irrelevant information and addressing inconsistencies. Imputation techniques are utilized to fill in missing data, while outliers—data points that deviate significantly from the rest of the distribution—are identified and addressed. If an outlier indicates inaccurate data, data scientists either remove or correct it. Additionally, incomplete and irrelevant data objects are eliminated.

K. Data anonymization

In certain cases, data scientists are required to remove or obfuscate attributes containing confidential information, especially when dealing with sensitive data from industries such as healthcare or banking.

L. Data sampling

When handling large datasets, data analysis can be time-consuming and require significant computational resources. To address this challenge, data scientists may utilize data sampling techniques to select a smaller yet representative subset of the data for model building and analysis. This approach allows for faster and more efficient analysis without compromising the accuracy of the results.

M. Data transformation

The final stage of preprocessing, known as feature engineering, is crucial for preparing data for machine learning or data mining applications. This stage might involve tasks like scaling or normalizing data and simplifying or merging attributes to ensure the data is ready for analysis and model development.

N. Scaling

Data often includes numerical attributes with a wide range of values, such as measurements in millimeters, meters, and kilometers. To bring uniformity to these data points, scaling is employed to adjust them to a common scale, typically between 0 and 1 or 1 and 10.

O. Decomposition

Handling complex features within a dataset can present challenges in pattern recognition. To address this, decomposition techniques are used to reduce complex, higher-order features into simpler, lower-level elements, and to derive new features from existing ones. This approach is particularly useful in time series analysis, such as when a market researcher needs to analyze monthly demand for air conditioners from data that initially represents quarterly demand.

P. Aggregation

Another key technique is aggregation, which involves merging multiple features into a single comprehensive feature. For example, aggregating customer age data into categories like 16-20, 21-30, and 31-40 years aids in demographic analysis and reduces the size of the dataset while retaining important information. Preparing and preprocessing data is a meticulous and prolonged process that requires choosing the right methods and making iterative modifications according to the business needs, along with considering the quality and volume of data available.

Q. Dataset splitting

For effective machine learning, it is advisable to segment the dataset into three distinct subsets: the training set, the validation set, and the test set.

R. Training set

The training set is used by data scientists to train and optimize the model, teaching it to extract necessary parameters from the data.

S. Test set

The performance and generalizability of a model are assessed using a test set, which consists of data that was not involved in the training process. This helps evaluate whether the model can apply its learning to new, unseen data and is crucial for mitigating overfitting—where a model is so finely tuned to the training data that it fails to perform well on any other data. Hence, using separate datasets for training and testing is vital for building robust models.

1) Validation set: The validation set plays a pivotal role in fine-tuning a model's hyperparameters, which are the overarching settings shaping the model's structure but are not learned from the data itself. These hyperparameters influence the complexity of the model and its pattern recognition capabilities. Generally, a dataset is partitioned with 80% used for training and the remaining 20% for testing. From the training segment, 20% is often allocated to the validation set. However, some experts suggest a distribution of 66% for training and 33% for testing, depending on the dataset's overall size.

T. Dataset-splitting

Enhancing the performance of a model can also be achieved by increasing the size of the training dataset. More testing data can improve the model's evaluation process, enhancing its ability to generalize and perform effectively across new datasets.

U. Modeling

At this stage, a data scientist trains multiple models to pinpoint the one that yields the most accurate predictions

1) Model training: After data preprocessing and segmentation into three parts, training begins on different models to see which one provides the most precise predictions. The training process involves supplying the training data to an algorithm, which then uses this data to build a model capable of predicting new outcomes, crucial for predictive analysis. The choice between supervised and unsupervised learning depends on whether the aim is to predict specific outcomes or to categorize data based on similarities.

V. Supervised learning

Supervised learning is utilized for data with known labels or outcomes, where attributes are pre-defined in historical data before training commences. This method enables a data scientist to tackle both classification and regression tasks effectively.

VII. Results

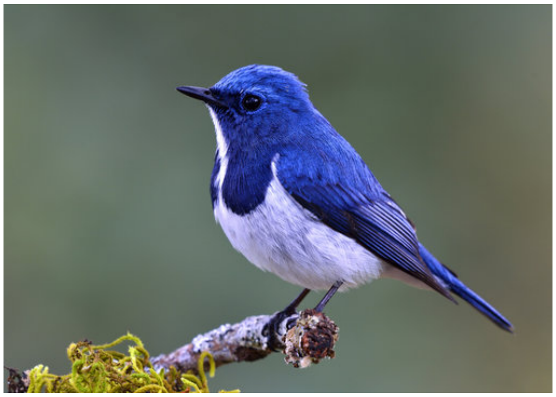

Fig. 4. Output:a bird sitting on a branch of a tree

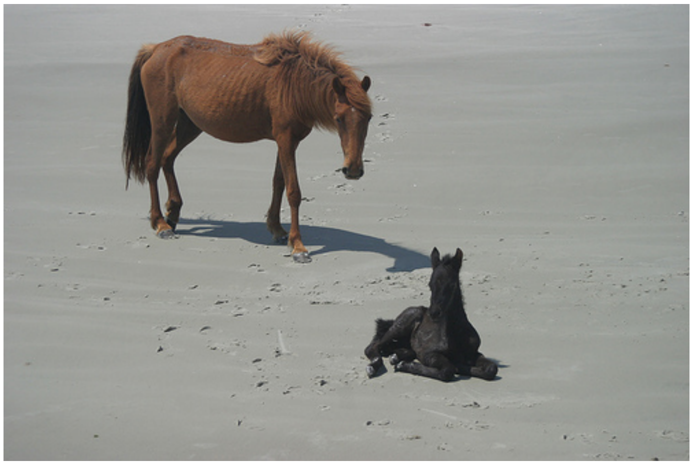

Fig. 5. Output: a couple of horses standing on top of a sandy beach

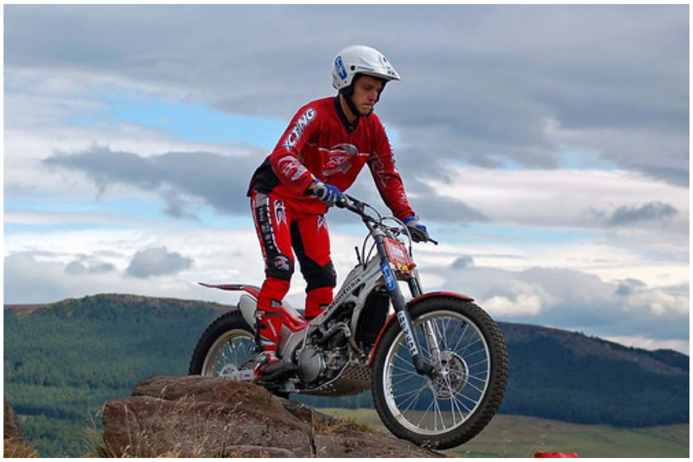

Fig. 6. Output: a man riding a motorcycle on a dirt road

References

[1] Graves, A.; Liwicki, M.; Fernández, S.; Bertolami, R.; Bunke, H.; Schmidhuber, J. (May 2009). "A Novel Connectionist System for Unconstrained Handwriting Recognition". IEEE Transactions on Pattern Analysis and Machine Intelligence.

[2] Li, Xiangang; Wu, Xihong (2014-10-15). "Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition

[3] Wu, Yonghui; Schuster, Mike; Chen, Zhifeng; Le, Quoc V.; Norouzi, Mohammad; Macherey, Wolfgang; Krikun, Maxim; Cao, Yuan; Gao, Qin (2016-09-26). "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

[4] Mayer, H.; Gomez, F.; Wierstra, D.; Nagy, I.; Knoll, A.; Schmidhuber, J. (October 2006). A System for Robotic Heart Surgery that Learns to Tie Knots Using Recurrent Neural Networks. 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 543–548.

[5] Rodriguez, Jesus (July 2, 2018). "The Science Behind OpenAI Five that just Produced One of the Greatest Breakthrough in the History of AI". Towards Data Science. Archived from the original on 2019-12-26. Retrieved 2019-01-15.

[6] Li, Xiangang; Wu, Xihong (2014-10-15). "Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition

[7] K. Cho, B. Van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder–decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empir-ical Methods in Natural Language

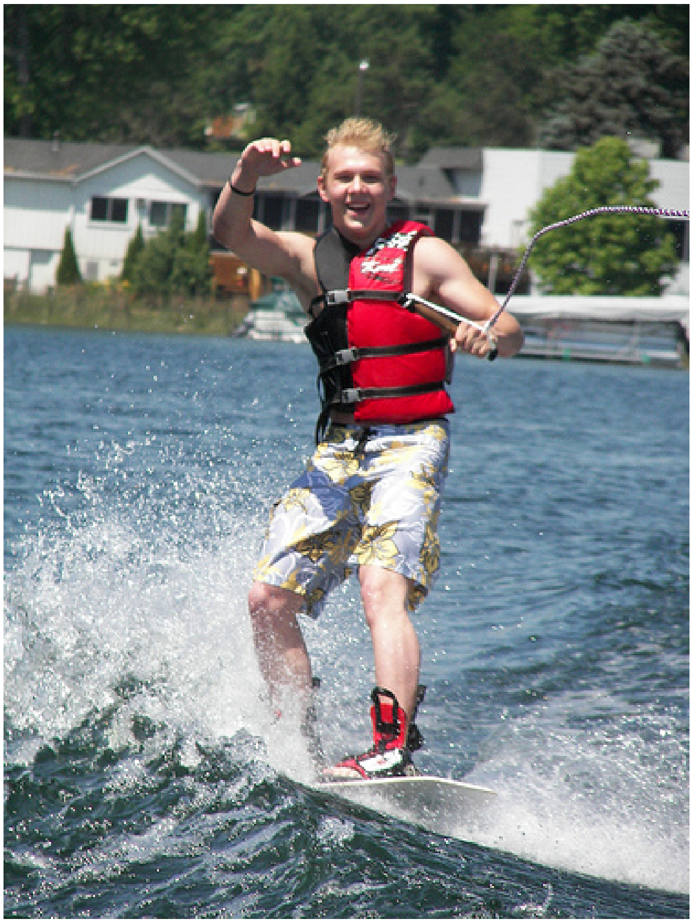

Fig. 7. Output: a man is surfing in the ocean with a surfboard

Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014.

[8] Rashtchian C, Young P, Hodosh M, Hockenmaier J. (2010) Collecting image annotations using Amazon's Mechanical Turk. In: Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's Mechanical Turk. Association for Computational Linguistics.

[9] D. Lin, An information-theoretic definition of similarity, in: Proceedings of the Fifteenth International Conference on Machine Learning

[10] Dewa Made Sri Arsa, I.P.A. Bayupati and Kadek Sastrawan ," Detection of fake news using deep learning CNN–RNN based methods"

[11] Anjali Samad,Bhagyanidhi and Vaibhav Gautam, "An Approach for Rainfall Prediction Using LSTM Neural Network.

[12] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate (2014), arXiv preprint arXiv: 1409.0473.

[13] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neu-ral networks, in: Advances in neural information processing systems, 2014, pp. 3104–3112 [4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Van- houcke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015

[14] H. Liu and P. Singh. ConceptNet - A practical common-sense reasoning toolkit. BT technology journal, 22(4):211–226, 2004.

[15] Young P, Lai A, Hodosh M, Hockenmaier J (2014) From image descriptions to visual denotations: new similarity metrics for semantic inference over event descriptions. Trans Assoc Comput Linguist 2:67–78