UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E INFORMÁTICA INDUSTRIAL

ANDREI DE SOUZA INÁCIO

CONTRIBUTIONS TO THE VIDEO CAPTIONING IN AN OPEN-WORLD SCENARIO USING DEEP LEARNING TECHNIQUES

TESE

CURITIBA
2023

ANDREI DE SOUZA INÁCIO

CONTRIBUTIONS TO THE VIDEO CAPTIONING IN AN OPEN-WORLD SCENARIO USING DEEP LEARNING TECHNIQUES

Contribuições para a descrição de vídeos em um cenário de mundo aberto utilizando técnicas de aprendizado profundo

Tese apresentado(a) como requisito para obtenção do título de Doutor em Ciências, do Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Heitor Silvério Lopes

CURITIBA

2023

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Curitiba

ANDREI DE SOUZA INACIO

CONTRIBUTIONS TO THE VIDEO CAPTIONING IN AN OPEN-WORLD SCENARIO USING DEEP LEARNING TECHNIQUES

Trabalho de pesquisa de doutorado apresentado como requisito para obtenção do título de Doutor Em Ciências da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Engenharia De Computação.

Data de aprovação: 28 de Agosto de 2023

Dr. Heitor Silverio Lopes, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Andre Eugenio Lazzaretti, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Helio Pedrini, Doutorado - Universidade Estadual de Campinas (Unicamp)

Dr. Manasses Ribeiro, Doutorado - Instituto Federal Catarinense

Dr. Pedro Henrique Bugatti, Doutorado - Universidade Tecnológica Federal do Paraná

ACKNOWLEDGEMENTS

I would like to begin by expressing my gratitude to my supervisor, Dr. Heitor Silvério Lopes, for his unwavering support, patience, guidance, and knowledge during my entire period at UTFPR. I have greatly benefited from his wealth of knowledge and invaluable experience.

I would also like to express my appreciation to my thesis committee members, Professors André Eugênio Lazzaretti, Hélio Pedrini, Manassés Ribeiro, and Pedro Henrique Bugatti, for their valuable insights and suggestions for this thesis.

I am grateful to my colleagues at LABIC for the conversations, suggestions, and constant help, especially when I lived in Curitiba.

I also want to extend my thanks to my parents and sisters for their guidance, love, and support throughout my life.

I also must express my very profound gratitude to my wife, Maria Eduarda, for her unwavering love, support, trust, patience, and understanding during my times of absence.

I thank all my friends and colleagues who cheered me on with each achievement during this journey and helped renew my energy in moments of doubt and uncertainty.

Finally, I would like to thank IFSC for believing in and investing in the education and training of its professors, as well as FUMDES for the financial support provided between 2021 and 2023. This work was carried out with the support of the UNIEDU/FUMDES postgraduate program.

RESUMO

INÁCIO, Andrei de Souza. **Contribuições para a descrição de vídeos em um cenário de mundo aberto utilizando técnicas de aprendizado profundo**. 2023. 150 f. Tese (Doutorado em Engenharia Elétrica e Informática Industrial) – Universidade Tecnológica Federal do Paraná. Curitiba, 2023.

A tarefa de descrição de vídeos representa um desafio significativo para as áreas de Visão Computacional e Inteligência Artificial, pois envolve a tradução do conteúdo visual de vídeos em linguagem natural. Apesar dos avanços significativos alcançados por meio de técnicas de aprendizado profundo, as abordagens existentes geralmente executam essa tarefa em um contexto de mundo fechado, presumindo que todas as ações e conceitos possíveis em uma cena, bem como o vocabulário, sejam conhecidos antecipadamente. No entanto, em aplicações do mundo real, novas ações e objetos podem surgir inesperadamente, exigindo novos vocabulários para descrever esses conceitos. Portanto, uma abordagem desejável para a descrição de vídeos em um ambiente de mundo aberto é aquela que pode descrever eventos conhecidos, detectar eventos desconhecidos e se adaptar incrementalmente para aprender a descrever esse conjunto de eventos desconhecidos, sem esquecer os eventos já aprendidos. Esta tese apresenta contribuições para o problema da descrição de vídeos em um cenário de mundo aberto. O primeiro método proposto é um sistema denominado OSVidCap, que visa descrever eventos conhecidos realizados por humanos em vídeos. O segundo método é uma abordagem de aprendizado incremental para a descrição de vídeos, permitindo a adaptação do modelo existente para aprender novas classes incrementalmente. Dois novos conjuntos de dados e um protocolo de avaliação foram criados para avaliar as abordagens de descrição de vídeo em um contexto de mundo aberto. Os resultados experimentais obtidos com estes conjuntos de dados demonstraram a eficácia dos métodos propostos.

Palavras-chave: Descrição de Vídeos. Aprendizado Profundo. Aprendizado Incremental. Visão Computacional. Aprendizado de Mundo Aberto.

ABSTRACT

INÁCIO, Andrei de Souza. **Contributions to the video captioning in an open world scenario using deep learning techniques**. 2023. 150 p. Thesis (PhD in Graduate Program in Electrical Engineering and Industrial Informatics) – Universidade Tecnológica Federal do Paraná. Curitiba, 2023.

Video captioning poses a significant challenge within the Computer Vision and Artificial Intelligence domains. It involves the challenging task of translating the visual content of videos into natural language descriptions. Despite significant advancements achieved through deep learning techniques, existing approaches usually perform such a task in a closed-world scenario, assuming all actions, concepts presented in a scene, and vocabulary are known in advance. However, new actions and objects may emerge unexpectedly in real-world applications, and new vocabulary may be necessary to describe those concepts. Therefore, an ideal video captioning approach for an open-world environment should be able to describe known events, detect unknown ones, and adapt incrementally to learn how to describe new events without forgetting what it has already learned. This thesis presents contributions to the video captioning problem in an open-world scenario. The first method, called OSVidCap, was proposed to describe concurrent known events performed by humans in videos and can deal with unknown ones. The second proposed method is an incremental learning approach for video captioning, designed to adapt an existing model to learn new events incrementally. Two novel datasets and a protocol for evaluating video captioning approaches in an open-world scenario are presented. Experimental results conducted on these datasets demonstrate the effectiveness of the proposed methods.

Keywords: Video Captioning. Deep Learning. Incremental Learning. Computer Vision. Open World Learning.

LIST OF ALGORITHMS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ANN	Artificial Neural Network
AVS	Ascending Vocabulary Size
BERT	Bidirectional Encoder Representations from Transformers
BERTScore	Bidirectional Encoder Representations from Transformers Score
Bi-SST	Bidirection Single-Stream Temporal
BLEU	Bilingual Evaluation Understudy
BPTT	Backpropagation Through Time
CIDEr	Consensus-based Image Description Evaluation
CLIP	Contrastive Language-Image Pre-training
CLIPScore	Contrastive Language-Image Pre-training Score
CNN	Convolutional Neural Networks
COCO	Common Objects in Context
CV	Computer Vision
DAP	Deep Action Proposals
DL	Deep Learning
DVS	Descending Vocabulary Size
ECCV	European Conference on Computer Vision
ELM	External Language Model
EMScore	Embedding Matching-based Score
ESGN	Event Sequence Generation Network
EVM	Extreme Value Machine
EVT	Extreme Value Theorem
EWC	Elastic Weight Consolidation
FAIEr	Fidelity and Adequacy ensured Image caption Evaluation metric
FCN	Fully Convolutional Network
FF	Feature Fusion
FNN	Feedforward Neural Network
GCN	Graph Convolutional Network
GNN	Graph Neural Network
GPaS	Graph-based Partition-and-Summarization
GPU	Graphics Processing Units
GRU	Gated Recurrent Unit
HAR	Human Action Recognition

HARL	Human Activities Recognition and Localization
HAT	Hard Attention to the Task
HWCM	Head Word Chain Matches
I3D	Inflated 3D Convnet
ICCV	International Conference on Computer Vision
ICPR	International Conference on Pattern Recognition
IL	Incremental Learning
KK	Known-Known
KU	Known-Unknown
LCEval	Learned Composite Metric for Caption Evaluation
LCS	Longest Common Subsequences
LEIC	Learning to Evaluate Image Captioning
LIRIS	Laboratoire d'InfoRmatique en Image et Systèmes d'information
LSMDC	Large Scale Movie Description Challenge
LSTM	Long Short-Term Memory
LwF	Learning without Forgetting
M-VAD	Montreal Video Annotation Dataset
METEOR	Metric for Evaluation of Translation with Explicit ORdering
MIMA	Model-Integrated Meta-Analysis
ML	Machine Learning
MLP	Multi-Layer Perceptron
MOWE	Mean of Word Embeddings
MPII-MD	MPII Movie Description
MSR-VTT	MSR Video to Text
MSVD	Microsoft Research Video Description Corpus
NACF	Non-Autoregressive Coarse to-Fine
NLP	Natural Language Processing
NMT	Neural Machine Translation
NNEval	Neural Network based Evaluation Metric
OSVidCap	Open-Set Video Captioning
Rec-LwF	Recurrent Learning without Forgetting
ReLU	Rectified Linear Unit
REO	Relevance, Extraness, Omission
RNN	Recurrent Neural Networks
RoBERTa	Robustly optimized BERT approach
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
RRS	Region Rank Similarity
SCN	Sequential Captioning Network
SGN	Semantic Grouping Network

SMURF	SeMantic and linguistic UndeRstanding Fusion
SPARCS	Semantic Proposal Alikeness Rating using Concept Similarity
SPICE	Semantic Propositional Image Caption Evaluation
SPURTS	Stochastic Process Understanding Rating using Typical Sets
SST	Single-Stream Temporal
TAMoE	Topic-Aware Mixture of Experts
tanh	hyperbolic tangent
TAP	Temporal Action Proposal
TDL	Target Detection and Localization
TEP	Temporal Event Proposal
TF-IDF	Term Frequency-Inverse Document Frequency
TI3D	Triplet Inflated 3D Neural Network
TIGEr	Text-to-Image Grounding based metric for image caption Evaluation
TRL	Teacher-Recommended Learning
TVT	Two-View Transformer
UK	Unknown-Known
UMIC	Unreferenced Metric for Image Captioning
UNINTER	UNiversal Image-TExt Representation learning
UU	Unknown-Unknown
VIFIDEL	VIsual Fidelity for Image Description EvaLuation
ViLBERT	Vision-and-Language BERT
ViLBERTScore	Vision-and-Language BERT Score
VLAD	Vector of Locally Aggregated Descriptors
WDS	Weight Distribution Similarity
WMD	Word Mover's Distance

CONTENTS

1 INTRODUCTION

The high availability of low-cost image and video acquisition equipment has significantly changed people's lives. Digital videos and images play an essential role in our society by facilitating communication and sharing information. Besides, such visual content has revolutionized various areas, including surveillance, environmental monitoring, health, business, and education.

Understanding and describing the visual content of images and videos in natural language is a challenging task in Computer Vision (CV). For this purpose, sophisticated techniques are required to process the diversity of human and object appearances in different environments and their relationships over time.

Video events are high-level semantic concepts humans perceive in a video sequence (LAVEE *et al.*, 2009). Each concept consists of an entity (human, object, action, or scene attributes) that occupies a specific position in a frame and may vary in size, color, and shape. Moreover, these visual contents can perform different actions and be described at different granularities and abstraction levels.

The video description task (or video captioning) has become a hot topic in CV. An approach that accurately describes video events may be used in various applications such as human-robot interaction, video indexing, assistance to the visually impaired, sign language understanding, and general-purpose video surveillance (AAFAQ *et al.*, 2019).

Current deep learning techniques have achieved state-of-the-art performance in several CV problems. They have effectively learned discriminative spatiotemporal features from raw data to solve several complex tasks, such as object detection and classification (BOCHKOVSKIY *et al.*, 2020; LIU *et al.*, 2020; DIWAN *et al.*, 2022), human action recognition (KONG; FU, 2022; ZHANG *et al.*, 2019; SINGH; VISHWAKARMA, 2019; GUTOSKI *et al.*, 2021a), video summarization (SOBUE *et al.*, 2019), semantic image segmentation (INÁCIO; LOPES, 2020), and video understanding (BUDVYTIS *et al.*, 2019). However, a step beyond the simple categorical classification of objects and actions in scenes is to describe events in human-comprehensible language.

The video description task requires solving many problems simultaneously, including object detection and classification, Human Action Recognition (HAR), visual relationships between humans and objects, and Natural Language Processing (NLP). Many studies have

recently achieved impressive results in the video captioning task using Deep Learning (DL) techniques (ULLAH; MOHANTA, 2022; YAN *et al.*, 2022a; ISLAM *et al.*, 2021; AMIRIAN *et al.*, 2020; AAFAQ *et al.*, 2019).

Despite the efforts and progress, the video description task is still an open problem. Existing approaches using DL methods are limited to a fixed list of activities provided in the training corpus. They have focused on generating a holistic description of short-length videos with only one primary action happening in the video. However, videos may have concurrent activities in practical applications, such as safety monitoring and surveillance. Humans can perform many different actions and create new movements and hand gestures.

A more realistic approach is to assume an open-set assumption for describing activities, which can adequately describe known events and deal with unknown ones. It is worth mentioning that the concept of "unknown class" came from open-set recognition, and it is different from the "unseen class", which came from zero-shot learning. The "unknown class" indicates a class without any information about it, and there are neither training instances nor other side information, including labels in the testing set. The "unseen class" denotes the class with no instances available in the training set but with semantic information available about it (GENG *et al.*, 2020).

The scenario mentioned above becomes even more challenging due to the lack of knowledge about unknown classes during training (OZA; PATEL, 2019). Therefore, in addition to detecting unknown events, some approach is necessary to incrementally learn how to describe this set of unknown events without forgetting the events already learned.

The ability of DL models to update or increment their capabilities when faced with new data is called incremental learning, and it is also an open problem (BELOUADAH *et al.*, 2021). In such approaches, the main challenge is the catastrophic forgetting problem (McCLOSKEY; COHEN, 1989), which is the performance deterioration of previously learned knowledge while learning new data. Such performance deterioration is a consequence of the stability–plasticity dilemma suffered by neural networks, which consists of a trade-off between learning new information (plasticity) and maintaining old knowledge (stability) (MERMILLOD *et al.*, 2013).

The main focus of this research is to provide some theoretical and methodological contributions to overcome these challenges in the video description in an open-world scenario. The problem definition and goals that guided this thesis are presented in the following sections.

1.1 PROBLEM DEFINITION

The central problem addressed in this work involves analyzing and understanding complex visual data from videos and generating a natural language sentence that describes a given video event within an open-world scenario.

DL methods have achieved state-of-the-art results in the video description task due to the availability of large datasets. Most methods are trained in a supervised manner and rely on the closed-set assumption. The vocabulary used to describe videos is obtained from the descriptions used during the training step. Thus, videos with actions and concepts never seen during training could not be adequately described. When faced with unknown events, such approaches usually produce hallucinations, such as sentences describing concepts and actions that do not appear in the video.

The open-set setting assumes the co-existence of known and unknown classes. However, new things always appear, as well as novel classes may appear. The open-world scenario introduces a continual learning paradigm that extends the open-set condition by assuming that new semantic classes are gradually introduced at each incremental time step. Such approaches in an open-world scenario have to detect unseen classes during the inference step and also learn incrementally based on the new classes in the old model (CHEN; LIU, 2018).

Class-incremental learning is a paradigm designed for environments where new classes may emerge at any time (ZHU *et al.*, 2017). Unlike the classification task, in which the output consists of a set of disjoint labels to be predicted, the output in the video captioning task is a sequence of words.

In this context, video captioning approaches within the open-world scenario must effectively detect unseen classes during inference and adapt incrementally by incorporating new classes into the existing model. This strategy is crucial to ensure accurate descriptions of events, avoiding the generation of inaccurate captions when facing unknown events. Furthermore, it is essential to expand the vocabulary used by the models with new words to adequately describe novel events and concepts.

One aspect that makes the video description task in the open-world context more challenging than the classification task is the overlapping vocabulary used to describe distinct video scenes. Different events usually contain a common vocabulary (articles, prepositions, nouns, and verbs). Furthermore, the presence of synonyms and homonyms can affect training. This

nature makes the task of class-incremental learning in video captioning even more challenging and complex.

Hence, the following questions arose during the research process: “Can a video description framework be designed to describe in natural language concurrent known video events in different contexts and deal with the unknown ones? What strategies could be employed to enhance the framework’s ability to learn and describe unknown events?” To answer these questions, the following hypothesis is proposed: Deep Learning techniques can detect and recognize video events, learn relevant features, and generate a sequence of words that can semantically describe concurrent events in videos in an open-set scenario. Furthermore, video captioning approaches can employ incremental learning to acquire the capability to describe novel events over time progressively.

1.2 OBJECTIVES

The general objective of this work is to propose methods for generating natural language descriptions of videos within the context of an open-world scenario. To clarify the approaches and contributions of this thesis, the general objective can be divided into more specific ones:

1. To propose a method to describe, in natural language, single and concurrent known events occurring in videos.
2. To investigate and devise a method to detect and recognize unseen and unknown events.
3. To investigate and devise a method to incrementally learn how to describe the unknown events detected.
4. To create new datasets for training the proposed models.
5. To validate the proposed approaches with public datasets.

1.3 DOCUMENT STRUCTURE

This Thesis is organized as follows. Chapter 2 presents theoretical aspects of video processing and DL methods, including the related works found in recent literature. Chapter 3 describes in detail the proposed methods. Chapter 4 reports the experiments, results obtained, and their respective discussion. Finally, Chapter 5 presents general conclusions and considerations

about the study and proposals for future work. Moreover, Appendix A presents a list of related works analyzed during the thesis period.

2 THEORETICAL ASPECTS AND THE STATE OF THE ART

This chapter presents the theoretical aspects of the leading models used and developed in this work. First, a brief introduction about Computer Vision (CV), Open-World Machine Learning, Deep Learning (DL), Incremental Learning, and Natural Language Processing (NLP) are presented, contextualizing them. Finally, a systematic mapping study regarding the works related to this thesis is presented.

2.1 COMPUTER VISION AND VIDEO PROCESSING

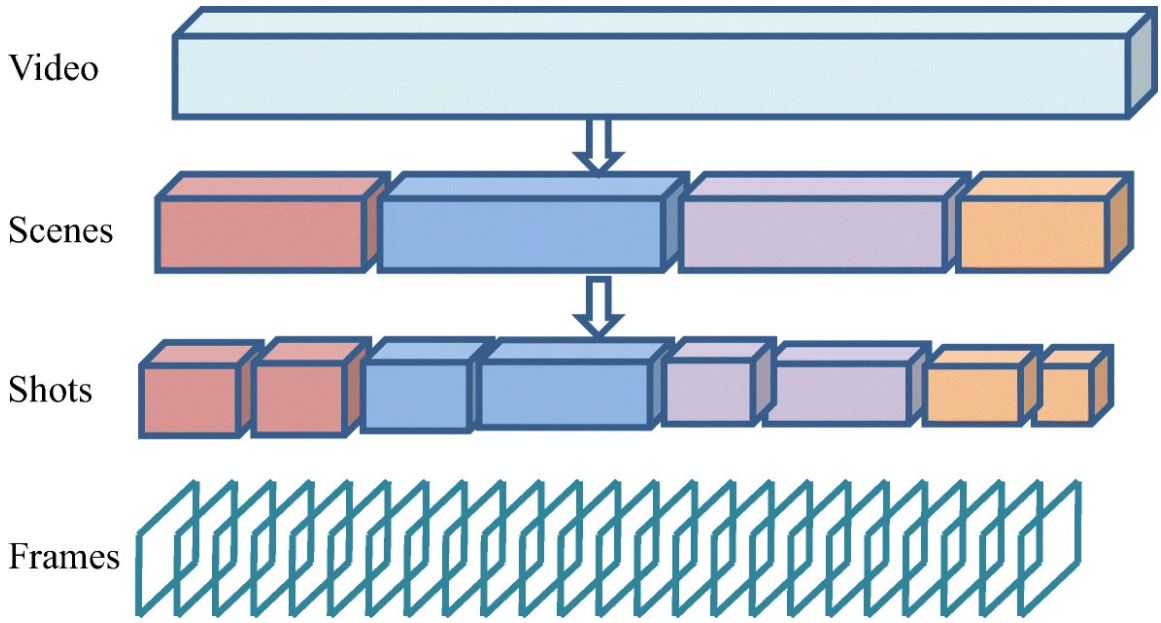
CV is a research field concerned with creating machines to see and understand the world from digital images and videos (BRUNELLI, 2009). CV can also be understood as the host of techniques to acquire, process, analyze, and understand complex higher-dimensional data, such as images and videos (JAHNE; HAUßECKER, 2000).

Video processing remains a challenging task, as it involves considering spatial and temporal information to understand objects and activities (JELODAR *et al.*, 2019). A video consists in a sequence of scenes, wherein each scene consists of one or more shots. These shots, in turn, are composed of individual frames. Each frame comprises a matrix of pixels, which is the small element of an image. Figure 1 illustrates a hierarchical structure representing the logical organization of a video. Typically, a standard video camera provides a video rate of 30 frames per second, which may vary according to the application.

The lack of correspondence between the low-level information extracted from raw images or videos, which is merely an ensemble of pixels, and the high-level meaning associated with them is called the semantic gap (PERLIN; LOPES, 2015). In common sense, CV techniques are designed to narrow the semantic gap problem and process visual information automatically.

With the DL techniques recently introduced, several approaches have achieved the state of the art in many fields of study, including image classification, video classification, and action recognition (POUYANFAR *et al.*, 2018). The use of DL for video processing has also provided several solutions for real-world applications, including surveillance systems (HG; S, 2020), human action recognition (SAHOO; ARI, 2019), health monitoring (PRATI *et al.*, 2019), traffic monitoring (JAIN *et al.*, 2019), and many more.

Figure 1 – Video content hierarchy.



Source: Rashmi e Nagendraswamy (2021).

2.2 OPEN-WORLD MACHINE LEARNING

Traditional supervised machine learning follows a closed-world assumption, in which models are trained to select the most likely class from a closed set. However, given the dynamic and open nature of the real world, new classes may emerge unexpectedly. This phenomenon leads to the concept referred to as “open set” problem (SCHEIRER *et al.*, 2013). In this scenario, samples with classes that were not seen during training are presented in the testing phase.

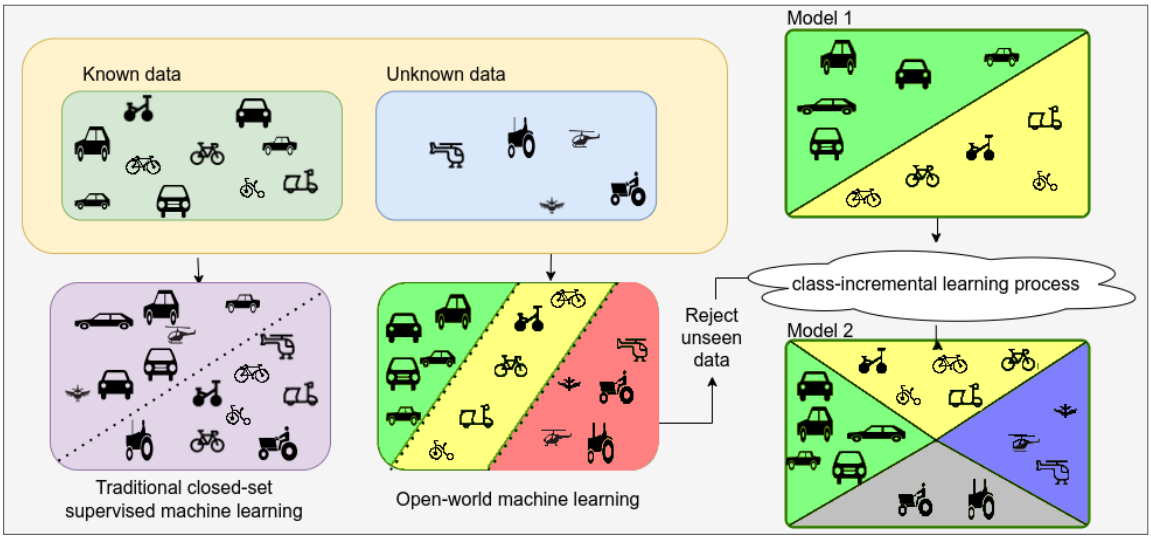
According to Geng *et al.* (2021), the recognition task should consider four categories of classes as follows:

- Known-Known (KK): classes with labeled training samples, and even have corresponding side-information, such as semantic or attribute information.
- Known-Unknown (KU): classes that are unknown to the classifier during training, but may appear during testing.
- Unknown-Known (UK): classes with no available samples in training but with semantic/attribute information available.
- Unknown-Unknown (UU): classes never seen in training and without any semantic-information during training.

Traditional models are classified within the KK category, operating under the assumption that both the training and testing data are known. In cases where the model explicitly incorporates "Other classes" or a detector trained with unclassified negative samples, it falls under the KU category. Zero-shot learning approaches predominantly concentrate on the UK category, aiming to recognize classes not faced during the training phase while leveraging the shared semantic information between KK and UK classes. Open set recognition encompasses a scenario where the model must accurately classify KK classes while simultaneously handling new classes not presented during training and lacking semantic information UU. In such instances, the classifier should be able to reject samples originating from the UU category.

A step beyond the open-set scenario, wherein the model only rejects samples from the UU category, lies the open-world scenario. In such a scenario, the model can detect and continuously learn new classes that emerge during testing (BENDALE; BOULT, 2015). Figure 2 illustrates a comparison between supervised machine learning and open-world machine learning. Traditional supervised machine learning involves using known instance labels during the training and inference stages. It follows the closed-set assumption, wherein the model assumes it knows all existing labels. When presented with unknown instances, it classifies them considering the known label categories. In the context of Open-world machine learning, the model can accurately classify instances belonging to the known classes while effectively rejecting the unknown ones. Such unknown instances can be labeled, and the model can be updated incrementally to learn these new classes.

Figure 2 – Comparison between traditional and open-world machine learning approaches



Source: Inspired by Parmar *et al.* (2023).

The open-world machine learning problem was formally defined by (BENDALE; BOULT, 2015; PARMAR *et al.*, 2023) as follows:

Let $\mathbb{Z}^+$ be the classes labeled by positive integers and $\Lambda_t \in \mathbb{Z}^+$ is the set of labels for known classes at time t. Let zero label (0) temporarily mark data as unknown. Therefore, $\mathbb{Z}$ includes both known and unknown labels. Let $x \in \mathbb{R}^r$ be the features (r is the dimension of x), and $f_y(x)$ is the recognition function; that is, if the $f_y(x) > 0$, then instances are marked as a known class and if $f_y(x) \leq 0$, then instances are marked as an unknown class, where $y \in \mathbb{Z}$. The solution to recognize any instance in an open-world scenario can be given as a tuple $[F,\phi,v,L,I]$ with:

- $F(x) : \mathbb{R}^r \rightarrow \mathbb{Z}$ as a multi-class open set recognition function that uses a vector function $\phi(x)$ of i per-class measurable recognition functions $f_i(x)$, and a novelty detector $v(\phi) : \mathbb{R}^i \rightarrow [0,1]$.

- $L(x) : \mathbb{R}^d \rightarrow \mathbb{N}^+$ is a labeling process applied to novel unknown data U_t from time t, resulting labeled data $D_t = \{(y_i,x_j)\}$ where $y_j = L(x_j) \forall x_j \in U_t$. Assuming the labeling finds m new classes, then the set of known classes becomes $\Lambda_{t+1} = \Lambda_t \cup \{i+1, ...i+m\}$.

- $I_t(\phi; D_t) : (F)^i \rightarrow (F)^{i+m}$ is an incremental learning function to scale, learn and add new recognition functions $f_{i+1}(x)...f_{i+m}(x)$.

Ideally, all these steps should be automated. However, in this thesis, the labels for the detected unknown classes are obtained through manual human labeling. The learning process in open-world machine learning can be defined into three steps (PARMAR *et al.*, 2023):

- **Step 1:** At time *t*, a multi-class classifier model M_t built by a learner on all previous classes $S^t = s_1, s_2, ..., s_t$. M_t can classify seen classes $s_i \in S^t$ or reject them as unseen classes and put them in a rejection set R_e. The R_e may have instances of more than one new or unknown class.

- **Step 2:** The system can identify the hidden classes c in R_e and prepare training sets from this data to find unknown classes.

- **Step 3:** The model M_t will learn from the updated training dataset and be updated to a new model M_{t+1}.

Many approaches have been proposed in recent years to tackle the open-world problem across various research fields, including computer vision, image processing, and natural language processing (PARMAR *et al.*, 2023). However, it is still an open problem due to the unpredictable nature of the upcoming events.

Apart from the open-world definition presented, a brief discussion of related topics found in Yang *et al.* (2021b) is introduced, which helps clarify the scope of this thesis:

- **Open-set scenario**: In such a problem, the classifier is required to accurately classify test samples from known classes and reject test samples from unknown classes, both simultaneously.
- **Domain Adaptation/Domain Generalization**: In such a problem, also known as the domain shift problem, the distribution changes, and the classifier is expected to continue accurately classifying the same class set. The main difference between DA and DG is that while DA requires additional but few training samples from the target domain, DG exclusively considers the original source domain data for adaptation and generalization.
- **Open-set domain adaptation**: The traditional Domain Adaptation problem assumes that the source and target domains belong to the same set of classes. In contrast, the concept of open-set domain adaptation introduces the scenario where the source and target domains share only a subset of object classes, with the majority of samples in the target domain belonging to classes not represented in the source domain.
- **Out-of-distribution**: Refers to the problem where a model faces samples during inference that are significantly different from the dataset on which the model was trained.
- **Zero-shot learning**: The model is trained with known classes similar to a traditional closed-set assumption. However, the test set contains unknown classes, and it is expected that the model can classify not only the known classes but also the unknown test samples with the help of additional information, such as label relationships.

2.3 DEEP LEARNING

DL is a machine learning field that enables computers to learn from experience and understand the world in terms of a hierarchy of concepts (GOODFELLOW *et al.*, 2017). It is not a recent technology, but it was made possible and popularized by the advent of fast Graphics

Processing Units (GPU) that enables massively parallel computation (CARRIO *et al.*, 2017). DL algorithms are composed of multiple hierarchical hidden layers between the input and output layers that extract increasingly abstract features from the input data.

Over the years, several different deep learning architectures have been proposed (SCHMIDHUBER, 2015), such as Convolutional Neural Networks (CNN), Fully Convolutional Network (FCN), RNN, Long Short-Term Memory (LSTM). CNN is the most popular model, and achieves impressive performance in feature extraction. RNN is a natural choice for sequence modeling tasks. Some of these models are the foundations and inspirations for the conception of this work and are presented in the following subsections.

2.3.1 Artificial Neural Network

An Artificial Neural Network (ANN) consists of parallel, connected arithmetic units, called neurons, designed to store experimental knowledge and make it available. They are inspired in the human brain in two aspects: first, the knowledge is obtained from the environment through a learning process; second, connections between neurons, known as synaptic weights, are used to store the knowledge (HAYKIN, 1999).

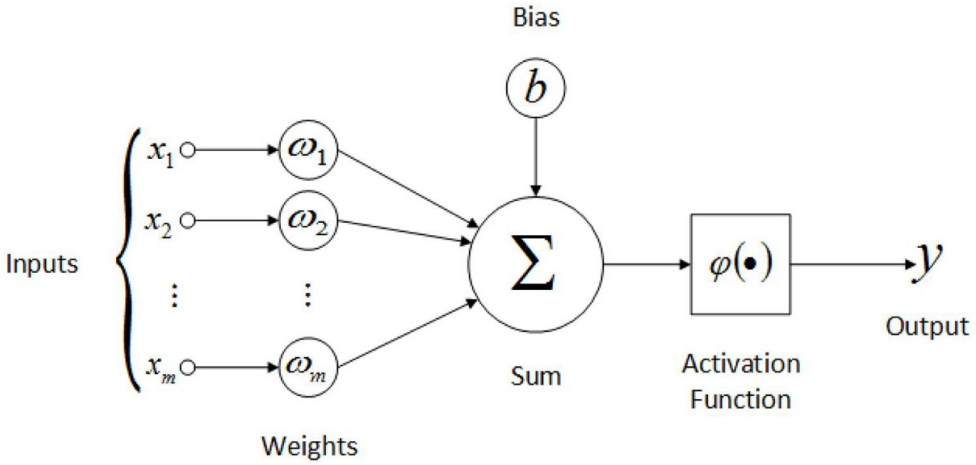
An ANN is composed of several neurons organized in layers. Each neuron computes a linear function (weighted sum of the input x) followed by an activation function. Optionally, a bias b can be added to adjust the output. Figure 3 depicts the basic computation of the neuron. The activation function is a non-linear function used to decide whether the information processed by a given neuron is relevant to the output or it should be ignored. The usual activation function used is the hyperbolic tangent (tanh).

The Feedforward Neural Network FNN, also called Multi-Layer Perceptron (MLP), is a class of ANN trained to map a given input $\mathbf{x}$ to a category y. It defines a mapping $\mathbf{y} = f(\boldsymbol{x};\boldsymbol{\theta})$ and learns the value of the parameter $\boldsymbol{\theta}$ that results in the best function approximation (GOODFELLOW *et al.*, 2017).

A typical FNN consists of neurons organized in an input layer, one or more hidden layers, and an output layer. Each neuron of a given layer can be connected to one or more neurons to the next layer. A fully connected layer is a layer in which all inputs are connected to every activation unit of the next hidden layer. Figure 4 presents an example of an FNN with two hidden layers.

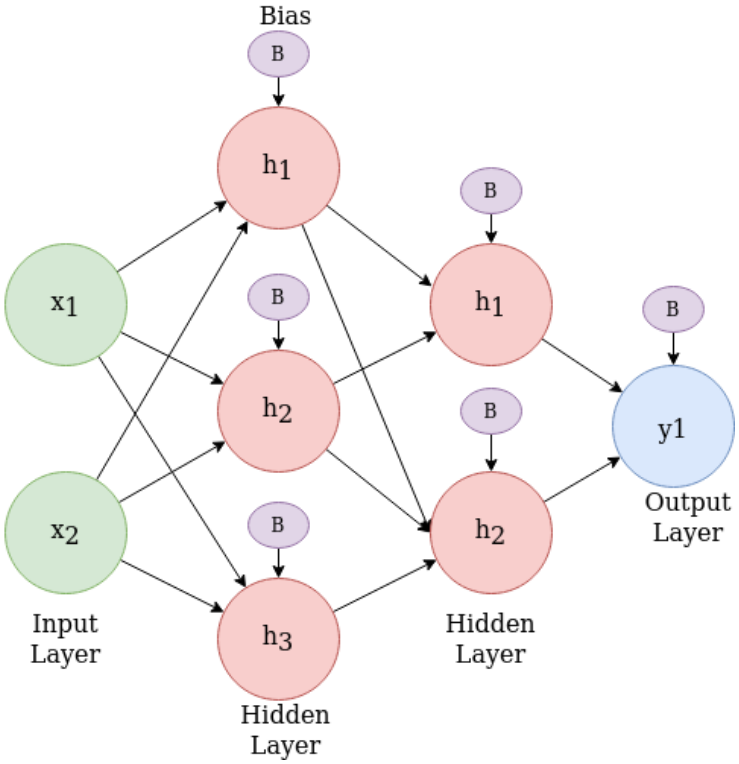
Before training, the network weights are often initialized with random values. The

Figure 3 – Non-linear model of a neuron.



Source: Haykin (1999).

Figure 4 – Example of an FNN with two hidden layers.



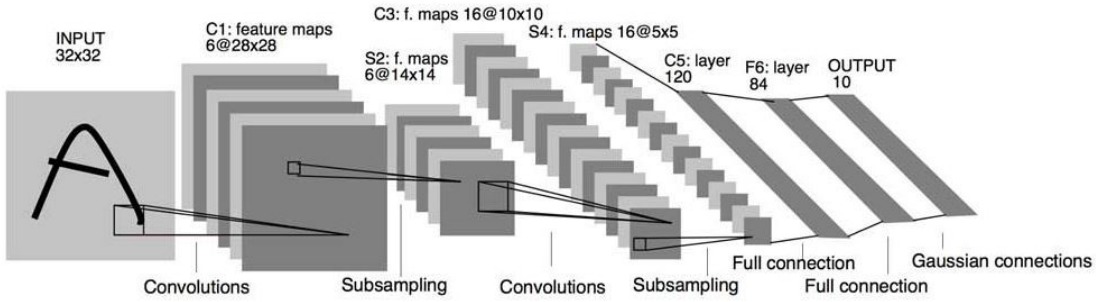
Source: Developed by the author.

training is then performed in three steps: First, a given input *x* flows forward through the hidden units, producing the output predicted *y*. Then, the error between the predicted and the expected result is calculated. Finally, the weights and bias adjustment are carried out using an iterative gradient descent-based optimization algorithm called backpropagation. These steps are performed several times to minimize the error between the predicted and the expected output.

2.3.2 Convolutional Neural Network

CNN is a type of FNN proposed by Fukushima (1980) and employs linear transformations called convolution layer, pooling layers, and fully connected layers to compute a hierarchy of features from the raw input data such as images or videos (LECUN *et al.*, 1998). They were popularized by Lecun *et al.* (1998) to achieve excellent performance in the task of handwritten digit image classification. The convolution and pooling layers work together to compute abstract features, and a fully connected layer is usually presented at the end of the architecture and works as a classifier. Figure 5 depicts an example of the CNN architecture.

Figure 5 – Architecture of LeNet-5 for digit recognition.

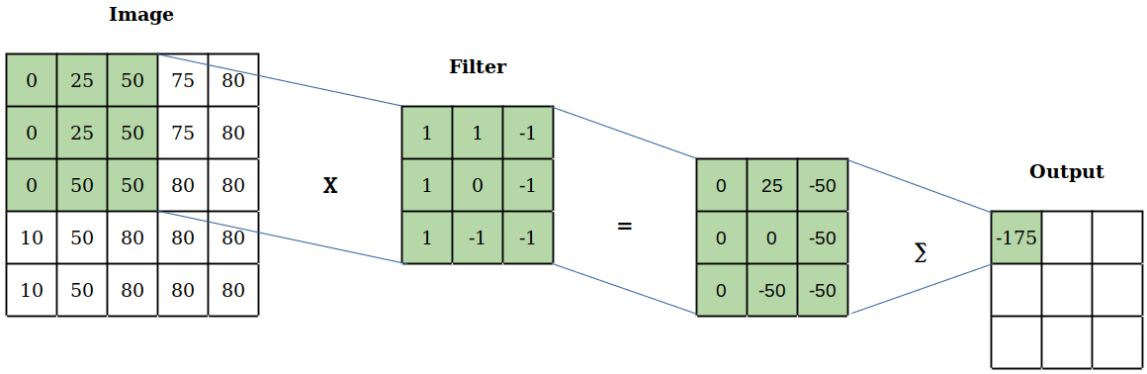


Source: Lecun *et al.* (1998).

The convolutional layer consists of a kernel, also called a filter, which is a small matrix trained on the data. The kernel slides over the pixels of the image and computes a weighted sum of the pixels, resulting in an output feature map. In addition to the kernel size, other hyperparameters that can be defined are the number of filters to be used in the convolution operation; the stride, which is the number of pixels the kernel will be translated at a time; and padding, which expands the input data artificially around the border to compute a feature map over all pixels and maintain the input shape. An activation function, usually the Rectified Linear Unit (ReLU) or *tanh*, is applied to the convolution output. Figure 6 shows an example of a 3×3 kernel applied to a 5×5 image.

A pooling layer consists of an operation used to modify the output dimensionality. It summarizes the outputs of neighboring groups of neurons in the same kernel map (KRIZHEVSKY *et al.*, 2012). It is used between convolutional layers to downsample the data representation and, consequently, the parameters learned. Similar to the convolution layer, the pooling is performed by a small window size that slides over the data to select the maximum value. The hyperparameters defined in this operation are the window size, stride, and padding. This operation is

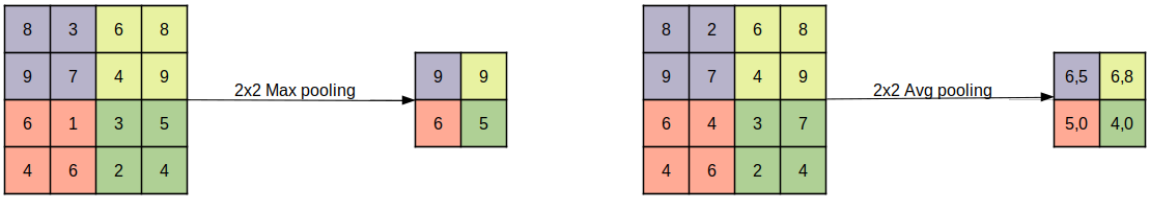
Figure 6 – Example of neural network convolution operation.



Source: Developed by the author.

known as max-pooling. There are other variants of pooling, such as min-pooling, which selects the minimum value in a given pooling step, and average-pooling, which computes the pixel value average presented in the windows in each pooling step. Figure 7 depicts an example of max and average polling.

Figure 7 – Example of neural network pooling operation.



Source: Developed by the author.

The fully connected layer takes the result of the last convolution or pooling process and performs a classification decision. The output layer contains a neuron for each possible label and outputs a vector representing the class probabilities for each label.

A serious problem faced when training a CNN is overfitting, a phenomenon that occurs in a model that learns too well on the training data, including noise and specific peculiarities, rather than finding a general predictive rule (DIETTERICH, 1995).

To overcome this problem, dropout (SRIVASTAVA *et al.*, 2014) is usually employed. It consists of dropping randomly some neurons during the training step, according to a pre-defined probability. This technique prevents the co-adaptation of neurons and improves the performance and generalization of the model.

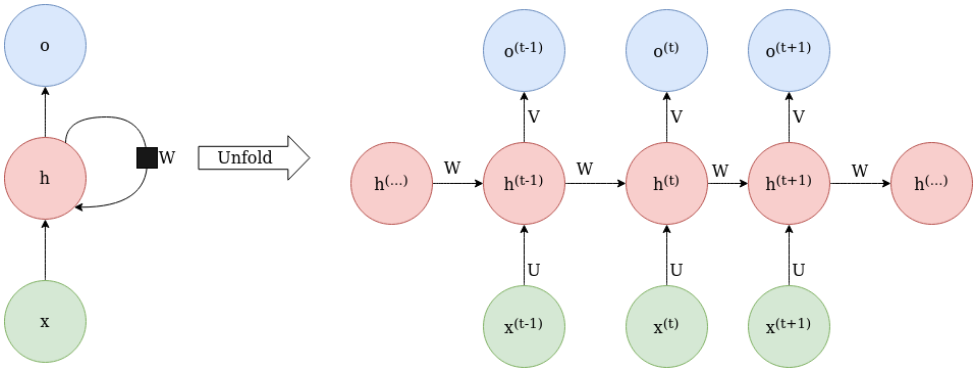
L1 and L2 regularization are also techniques used to prevent overfitting (NG, 2004). They consist of a regularization term added to the cost function that penalizes large weights/parameters.

The data augmentation technique suggested by Krizhevsky *et al.* (2012) can also be used to avoid overfitting. It consists of artificially enlarging the dataset using transformations to produce new images from the original images.

2.3.3 Recurrent Neural Network

RNN is a family of neural networks for processing sequential data (GOODFELLOW *et al.*, 2017). This architecture has the ability to selectively pass information across sequence steps while processing sequential data one element at a time, allowing to model input and/or output consisting of sequences of elements that are not independent (LIPTON *et al.*, 2015). A basic architecture of the RNN, also called Vanilla RNN, is depicted in Figure 8.

Figure 8 – Example of RNN architecture that maps an input sequence of x values to a corresponding sequence of output o values.



Source: Developed by the author.

Given a sequence of inputs (x^1, x^2, x^3, ..., x^T), a standard RNN computes a sequence of outputs (y^1, y^2, y^3, ..., y^T) by iterating the following equations.

$$h^{(t)} = \tanh(Wh^{(t-1)} + Ux^{(t)} + b) \tag{1}$$

$$o^{(t)} = Vh^{(t)} + c \tag{2}$$

$$y = softmax(o^{(t)}) \tag{3}$$

where *W*, *U* and *V* are weight matrices, *b* and *c* are bias vectors, $h^{(t-1)}$ is the previous state and $x^{(t)}$ is the input at the instant t.

The RNN training is performed using the Backpropagation Through Time (BPTT) algorithm, which consists of performing a forward propagation pass moving left to right through

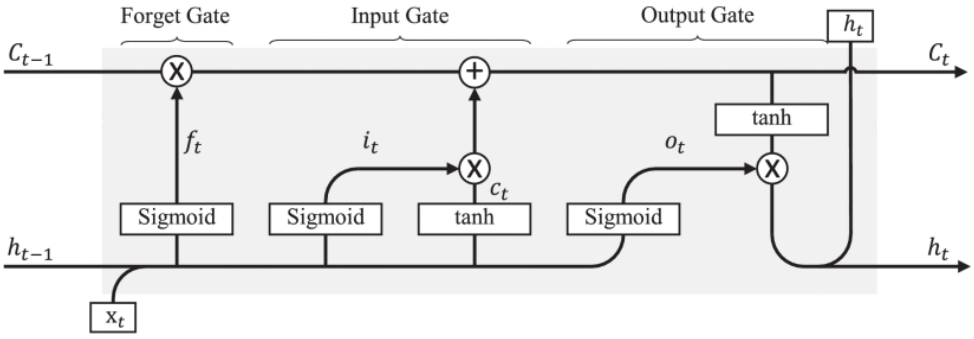
time, followed by a backward propagation pass moving right to the left. The final loss is computed by performing the sum of the losses over all the time steps.

Despite being suitable for many applications, including sequence recognition and time-series prediction, RNN faced difficulties in performing tasks with long-term dependencies (BENGIO *et al.*, 1994). The well-known exploding and vanishing gradient problems happen during the training step when the gradients are propagated back in time to the initial layer. To overcome these problems, many variants of the RNN have been proposed. The most popular approaches are LSTM and Gated Recurrent Unit (GRU).

2.3.4 Long Short-Term Memory

The LSTM (HOCHREITER; SCHMIDHUBER, 1997; GERS *et al.*, 1999) architecture consists of a set of memory blocks. Each block contains a memory cell and three gate units responsible for memory manipulation. The input gate provides a way to add information into the state cell. The forget gate provides a way to reset the cell and thus ignore previous information. When activated, the output gate decides which information will be made available to the output and the hidden state of the block. The memory block structure of the LSTM is illustrated in Figure 9.

Figure 9 – Example of LSTM architecture that maps an input sequence of x values to a corresponding sequence of output o values.



Source: Yu *et al.* (2017).

The detailed calculations of the LSTM model is presented as follows:

$$h_t = \tanh(C_t) * o_t \tag{4}$$

$$C_t = \sigma(f_t * C_{t-1} + i_t * c_t) \tag{5}$$

$$c_t = \tanh(x_t U^g + h_{t-1} W^g) \tag{6}$$

where U^g and W^g are weight matrices, x_t is the input at time t, h_{t-1} is the previous state, and f_t, i_t, and o_t are the forget, input and output gates, respectively.

The detailed calculations of unit gates are as follows:

$$f_t = \sigma(x_t U^f + h_{t-1} W^f + b_f) \tag{7}$$

$$i_t = \sigma(x_t U^i + h_{t-1} W^i + b_i) \tag{8}$$

$$o_t = \sigma(x_t U^o + h_{t-1} W^o + b_o) \tag{9}$$

where $U^f, U^i, U^o, W^f, W^i, W^o$ are weight matrices, b_f, b_i, b_o are bias vector, and σ denotes the sigmoid activation function. The LSTM has been shown extremely successful in many applications, such as Action Recognition, Sentiment Analysis, Video Description, and others.

2.4 INCREMENTAL LEARNING

Traditional supervised Machine Learning (ML) techniques assume that all the data needed to train the models is available during the training stage. After training, such models are deployed for inference with fixed parameters and take the assumption that the data distribution will not change. However, in real applications, the data distribution may change over time, which is referred to as the concept drift phenomenon (LU *et al.*, 2019b).

Depending on the study hypothesis, different terms have been used to refer to studies related to incremental learning, and using such terms is not always consistent (GEPPERTH; HAMMER, 2016). Incremental Learning (IL), also called lifelong learning or continuous learning, refers to an ML paradigm where new data is used in the learning process while the system is operating (CHEN; LIU, 2018).

Such a strategy may be required for several reasons, including the high computational cost of obtaining a large number of samples before the learning process; the learning algorithm can not deal if directly applied to all the available training data; new examples become available over time; the data itself is time-dependent; deal with nonstationary distribution, adapting to the changes in underlying data distributions (concept drift) (GENG; SMITH-MILES, 2009).

The objective of IL is to train a model from a set of tasks incrementally, where each task has data with associated labels. According to van de VEN *et al.* (2022), the supervised IL methods can be categorized into three scenarios:

- **Task-Incremental Learning**. Each task may contain data from a different set of classes, and the algorithm knows (in both the training and testing stages) which task must be performed.
- **Domain-incremental Learning**. Each task contains data from classes presented in previous tasks. The structure of the algorithm is always the same for all tasks. Only the input distribution data changes over the tasks.
- **Class-incremental Learning**. Each task contains data from different classes, and the algorithm has to incrementally learn to discriminate a growing number of classes over time. In this scenario, the task identifier is not provided, and the model must be able to distinguish between all classes from all tasks during inference time.

According to (MASANA *et al.*, 2023), the class-incremental learning problem can be formally defined as follows:

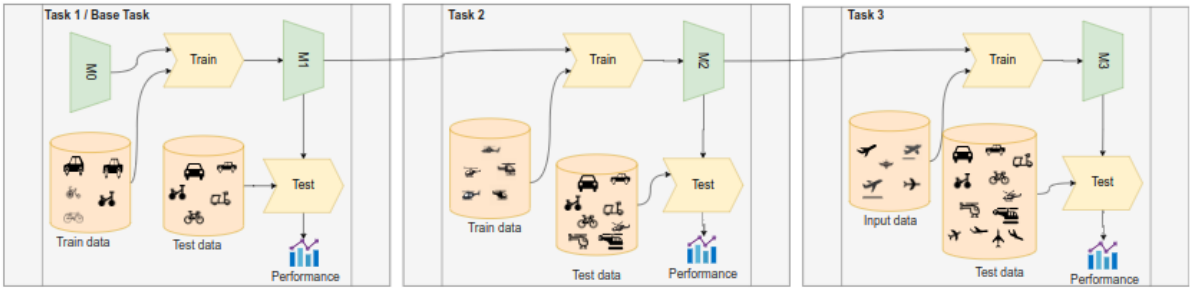
Let $T = [(C^1, D^1), (C^2, D^2), (C^3, D^3), ..., (C^n, D^n)]$ be a set of pairs, where each pair (C^t, D^t) represents a task t. In each task t, $C^t = \{c_1^t, c_2^t, ..., c_{n^t}^t\}$ is a set of classes and D^i is the training data for the that task t.

At each incremental learning step, a model M_t is trained using the data D_t. After, the performance of model M_t is evaluated from the union of all previous tasks up to task t. This union is denoted as $\cup_{i=1}^{t} C^i$.

Figure 10 depicts an example of a class-incremental learning classifier model. First, the model base is trained to classify cars and motorcycles (Task 1). Then, the resulting model is updated to learn helicopters (Task 2). Finally, the model is also updated to learn airplanes (Task 3). Notice that, in a class-incremental learning scenario, the architecture must be incremented to incorporate the class of new tasks, and the last trained model (Task 3) should classify all classes trained in previous tasks.

In the Deep Learning context, many researchers have studied the class-incremental learning problem (MASANA *et al.*, 2023). The main challenge is dealing with the catastrophic forgetting problem (McCLOSKEY; COHEN, 1989; RATCLIFF, 1990). Such a phenomenon is characterized by the abrupt loss of knowledge about previous tasks after training the model to

Figure 10 – Example of class-incremental learning system.



Source: Developed by the author.

predict a new task. The weights in the network importance for previous tasks are changed to meet the objectives of task B (KIRKPATRICK *et al.*, 2017). In addition, class-IL approaches face the challenge of learning and distinguishing new classes from previously learned ones.

2.4.1 Class-incremental learning methods

Different algorithms have been proposed for the Incremental Learning task and can be categorized in three families (De LANGE *et al.*, 2022) as summarized below.

Replay Methods. This family of methods primarily relies on utilizing a portion of data or features from previous tasks to mitigate the issue of catastrophic forgetting. These approaches typically involve storing a subset of exemplars per class, which are then reused as inputs for rehearsal or employed to constrain the optimization of the loss function for the new task. This helps prevent interference from previous tasks and allows retaining previously learned information. The main drawback of such methods is storing sample data in memory to represent previous distribution data, which demands extra computation resources.

Regularization-based methods. This family of methods includes regularization terms in the loss function to mitigate catastrophic forgetting while learning new data. Several approaches have been proposed in this category, such as Elastic Weight Consolidation (EWC) (KIRKPATRICK *et al.*, 2017), and Learning without Forgetting (LwF) (LI; HOIEM, 2018). The main difference among them is how they estimate the parameter importance in the model to calculate the penalty included in the loss function.

Parameter isolation methods. This family of methods explores techniques to isolate and freeze network parameters partially, trying to avoid forgetting previous tasks. In such a strategy, the network model may be fixed or with dynamic architecture by increasing the depth and complexity. An example of one approach that explores such techniques is the HAT(SERRA

et al., 2018). It employs an attention mechanism to learn how important the network weights are during a training session for a given task. Then, when training new tasks, the update of important network parameters of previous tasks can be constrained based on such attention masks.

2.5 THE EXTREME VALUE MACHINE

The Extreme Value Machine (EVM) was initially proposed by Rudd *et al.* (2018) to perform open-set classification, offering a robust solution to the challenges posed by classifying instances that do not belong to any predefined class. Its fundamental concept lies in modeling each class in the training set using a set of extreme vectors. These extreme vectors are associated with a Probability of Sample Inclusion (Ψ), which determines the likelihood of a sample belonging to a specific class.

The key concept of EVM is the use of margin distributions, which is the distribution of the half margin distances of the training data. In the original formulation, one can consider $\mathbf{x}_i$ as a training sample and y_i the corresponding label. Considering $\mathbf{x}_i$ and $\mathbf{x}_j$, where $\forall j, y_j \neq y_i$, $\mathbf{x}_j$ can be considered the nearest point to $\mathbf{x}_i$ and, in this case, the margin estimate for the pair $(\mathbf{x}_i, \mathbf{x}_j)$ is given by $\mathbf{m}_{ij} = || \mathbf{x}_i - \mathbf{x}_j || /2$.

The $\mathbf{m}_{ij}$ value can be computed for the τ nearest points and the distribution of the margins is estimated with those points using the Extreme Value Theorem (EVT). The EVT states that the minimum values of $\mathbf{x}_i$ is given by a Weibull distribution (RUDD *et al.*, 2018). The probability of inclusion Ψ for a point $\mathbf{x}'$ is given by

$$\Psi(\mathbf{x}_i, \mathbf{x}', \kappa_i, \lambda_i) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}'\|}{\lambda_i}\right)^{\kappa_i}, \tag{10}$$

in which $\|\mathbf{x}_i - \mathbf{x}'\|$ is the distance between $\mathbf{x}'$ and $\mathbf{x}_i$, λ_i and κ_i are the Weibull's shape and scale parameters.

Each Ψ is considered an EVT rejection model and $\Psi(\mathbf{x}_i, \mathbf{x}', \kappa_i, \lambda_i)$ corresponds to the probability that a sample is not beyond the negative margin. Even though a sample has zero probability around the margin, the model can also be extended to support soft margins. The probability that a point $\mathbf{x}'$ belongs to class C_l, where l is the class index, is given by Equation 11:

$$\hat{P}(C_l|\mathbf{x}') = \text{argmax}_{i:y_i=C_l} \Psi(\mathbf{x}_i, \mathbf{x}', \kappa_i, \lambda_i). \tag{11}$$

Finally, the classification function is:

$$y^* = \begin{cases} \text{argmax}_{i:y_i=C_l} \hat{P}(C_l|\mathbf{x}'), & \text{if } \hat{P}(C_l|\mathbf{x}') \geqslant \delta \\ \text{unknown}, & \text{otherwise} \end{cases}, \tag{12}$$

in which δ is a threshold responsible for defining the boundary between known and open-space.

According to Gutoski *et al.* (2021a), many redundant $[\mathbf{x}_i, \Psi(\mathbf{x}_i, \mathbf{x}', \kappa_i, \lambda_i)]$ pairs can be discarded with minimal impact on performance to reduce the size of the model. For further details about the EVM, including the mathematical foundations, can be found in the original paper by Rudd *et al.* (2018).

2.6 TRIPLET INFLATED 3D NEURAL NETWORK

The Triplet Inflated 3D Neural Network (TI3D) is a Deep Metric Learning Neural Network introduced by Gutoski *et al.* (2021a). It uses the Inflated 3D Convnet (I3D) as the base model to build a cosine triplet loss network. The TI3D learns a feature mapping such that intra-class distances are small and inter-class distances are large.

The TI3D takes three inputs: Anchor, Positive, and Negative. For the human action recognition task, the Anchor (a) represents a video of any given action, the Positive (p) represents a video of the same action, and the Negative (n) represents a video of a different action, both w.r.t. the anchor. Given N (a,p,n) triplets, the Triplet loss function L is defined by:

$$L_\Theta = \sum_{i=1}^{N} \Big[\Theta(f(\mathbf{x}_i^a),f(\mathbf{x}_i^p)) - \Theta(f(\mathbf{x}_i^a), f(\mathbf{x}_i^n)) + \alpha\Big]_+ . \tag{13}$$

in which i is the triplet index, $f(\mathbf{x}^a)$, $f(\mathbf{x}^p)$, $f(\mathbf{x}^n)$ are the Anchor, Positive and Negative embeddings, respectively, α is the margin parameter, and Θ denotes the cosine distance between two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$:

$$\Theta(\mathbf{x}_i,\mathbf{x}_j) = 1 - \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\| \mathbf{x}_i \| \| \mathbf{x}_j \|}. \tag{14}$$

Additionally, the symbol $+$ indicates the operator $max(\beta,0)$, for $\beta = \Theta(f(\mathbf{x}_i^a),f(\mathbf{x}_i^p)) - \Theta(f(\mathbf{x}_i^a), f(\mathbf{x}_i^n)) + \alpha$, which imposes $L_\Theta \geq 0$ for every $f(\mathbf{x}_i^a),f(\mathbf{x}_i^p)$ and $f(\mathbf{x}_i^a), f(\mathbf{x}_i^n)$ pairs, since $max(\beta,0) = 0, \forall\, \beta \in \mathbb{R} \mid \beta < 0$. This loss function attempts make the cosine distance between Anchor and Positive samples smaller than the distance between the Anchor and Negative instances by, at least, a margin of α. Alternatively, it will force examples of the same class to be mapped closer than examples of different classes (or even previously unknown examples).

2.7 NATURAL LANGUAGE PROCESSING

Natural Language Processing (NLP) is the subfield of Computer Science concerned with the use of computational techniques to learn, understand, and produce human language

content (HIRSCHBERG; MANNING, 2015).

Traditional NLP approaches focused on developing hand-coded rules and algorithms to process natural language (NG; ZELLE, 1997). Currently, most approaches for NLP are corpus-based learning based on statistical methods, or other automated learning techniques over corpora of natural language examples to automatically induce suitable language-processing models (OTTER *et al.*, 2021). NLP is employed in several real-world applications, including text classification, machine translation, chatbots and virtual assistants, sentiment analysis, and text generation.

The raw text used as input in natural language processing systems requires some preprocessing steps before it can be used for analysis. The most frequent methods used for preprocessing include:

- **Tokenization** consists of splitting the input text, which is a sequence of characters, into subunits called tokens. The tokenizer methods are usually based on rules, punctuation, or white spaces.
- **Stop words** consists of removing words considered irrelevant for a given approach. For example, in a classification problem, frequent words such as *a, the, after,* and *there* are unlikely to be useful.
- **Stemming** is a type of text normalization, in which the variant words forms (e.g. trouble, troubling, troubled) are reduced to their root form (troub).
- **Lemmatization** consists of converting words (e.g., was, meeting, cars) in their base or dictionary form (e.g., be, meet, car), known as the lemma.
- **Parts of speech tagging** consists of assigning parts of speech to each word, such as noun, verb, adjective, etc.

Similar to images, which can be mathematically modeled by analog or digital signals, text data also can be modeled as numbers to be automatically processed by computers (LI; YANG, 2018).

The one-hot embedding is the easiest way to represent a word in a discrete vector. In this method, each word has an index in the vocabulary set and is represented as a 1-D vector made up of zeros, with a 1 in the corresponding dimension for the word. The main drawbacks

faced by the one-hot embedding are the lack of semantic between words and a phenomenon called the curse of dimensionality, which is related to the large dimension size of the vectors.

Bag-of-Word (BoW) model is another way to extract text representation often used in NLP. It learns a vocabulary from all the sentences (or documents) and then models each sentence (or document) by computing the occurrence of words (DEEPU *et al.*, 2016). The BoW is commonly used in text classification tasks. The main drawback is that it does not take into account the word order or the similarity between different words.

Word embeddings are real-valued word representations able to capture lexical semantics and trained on natural language corpora (BAKAROV, 2018). Many techniques can be used to learn a word embedding from text data, including GLOVE (PENNINGTON *et al.*, 2014), Word2Vec (MIKOLOV *et al.*, 2013), fasttex (MIKOLOV *et al.*, 2018) and Embedding Layer, a learnable linear mapping that maps each word onto a low dimensional latent space. The main drawbacks of the word embeddings are that some cannot deal with unknown or out-of-vocabulary words and the lack of interpretability of the real values that make up the embedded vectors. Despite this, the pre-trained word embeddings available for free on the internet have proven to be invaluable for improving performance in natural language analysis tasks, which often suffer from a paucity of data (QI *et al.*, 2018).

2.8 STATE OF THE ART

Following the guidelines proposed by Kitchenham (2004), a systematic literature review was initially conducted to review and analyze the strategies developed for the video description problem. Such a review encompassed studies published between 2017 and 2020. Then, the most relevant studies were included in this literature review. The data sources, IEEE Xplore, ACM Digital Library, Springer Link, and Science Direct, were considered for this literature review due to their relevance in computing and vision computation. Besides, Google Scholar was also used to search for promising in-progress works or articles published outside the computing domain, which may significantly contribute to this work.

The papers were classified into three main topics, which are described in the following subsections, one presenting current state-of-the-art methods, one for the existing datasets used to train and evaluate those proposed models, and one for evaluation metrics employed in the video description task.

2.8.1 Video Description methods

Early proposed methods for the video description task started with template-based methods in which the Subject(S), Verb(V), and Object(O) were detected and then used in a sentence template (BARBU *et al.*, 2012; ROHRBACH *et al.*, 2013; KRISHNAMOORTHY *et al.*, 2013). Although these methods could generate descriptions based on grammar, they did not consider the spatial and temporal associations between entities. Moreover, the obtained output variety suffers because of the implicit limitation (AMARESH; CHITRAKALA, 2019). Inspired by the rapid development of DL techniques in the CV and NLP areas, video description research has become a hot topic. The video description approaches based on DL methods are mainly designed in the encoder-decoder architecture. The encoder usually combines 2D CNN, 3D CNN, and LSTM that converts the input video into a feature vector representation of fixed length. The decoder is usually an LSTM or GRU that generates a sequence of words.

Venugopalan *et al.* (2015b) proposed one of the first approaches based on deep neural networks for the video captioning task. A variant of AlexNet, pre-trained on a subset of the ImageNet dataset, was used to extract visual features from frames. Then, the mean pooling method was employed, resulting in a single vector representing the entire video. Finally, two stacked LSTM were used to generate the sentence.

The first end-to-end approach was proposed by Venugopalan *et al.* (2015a). The authors also used CNN to extract features from frames, but instead of performing a mean pooling in the features, they used an LSTM to consider salient and temporal features in the encoder. Similar to Venugopalan *et al.* (2015b), they also used an LSTM as a decoder to generate the sequence of words. These models performed better than the previous approaches at the time, and they could generate sequences effectively without any templates. However, their approaches did not adequately capture temporal information from the video, and important objects were not described. Also, their strategies could only generate descriptions for short clips with one action per clip.

Since then, several approaches based on the encoder-decoder framework using DL methods have been proposed. They commonly use pre-trained deep learning models, such as VGGNet (SIMONYAN; ZISSERMAN, 2015) or ResNet (HE *et al.*, 2016), to extract spatial features from frames. These features are usually combined across the frames by an average pooling or max-pooling operation, resulting in a single fixed-length feature vector representation

for a short video clip. Besides, the C3D (TRAN *et al.*, 2015) or I3D (CARREIRA; ZISSERMAN, 2017) models, pre-trained in large datasets such as the Sports-1M (KARPATHY *et al.*, 2014) or Kinetics (KAY *et al.*, 2017), are used to extract temporal features. Using pre-trained models on large datasets provides a strong visual representation of objects, actions, and scenes depicted in the video (VENUGOPALAN *et al.*, 2015b). The use of multimodal data, such as visual, audio, motion, and textual information, was also explored in some works (JIN *et al.*, 2016; DU *et al.*, 2019). Due to the higher improvement achieved by such approaches, multimodal data is often adopted nowadays.

Despite achieving satisfactory results, applying pooling methods to frames may discard useful information presented in specific frames. To tackle such issue, many approaches have been proposed to use attention mechanisms to dynamically select spatial and temporal features focusing on important frames and regions inside them, providing meaningful visual evidence for caption generation (TU *et al.*, 2017; SONG *et al.*, 2017; AAFAQ *et al.*, 2019; FRANCIS; HUET, 2019; YAN *et al.*, 2020; AHMED *et al.*, 2021; GHADERI *et al.*, 2022; JI *et al.*, 2022).

The attention mechanism has improved the video captioning task, suggesting that this method can improve descriptions by obtaining more representative and high-quality features for a given video. Although such approaches generate better descriptions using attention mechanisms, they can be misused and degrade overall visual caption performance when applied to non-visual words (e.g., conjunctions and articles). Most of the words generated by video captioning approaches are non-visual, and their relevance depends on the previously generated one. Thus, the use of attention mechanisms to align target words with corresponding visual content is exploited by some approaches (XIAO; SHI, 2018; GAO *et al.*, 2020a; TU *et al.*, 2021). They employ attention mechanisms to select visual features and video attributes (semantic concepts generated by a model previously trained) to decide whether the words generated at a time step depend on visual or contextual information.

Another issue when describing videos in natural language is dealing with edited videos, including movie ones. Such videos may have scene transitions that can be segmented into short scenes. Although temporally consistent, they have different appearances, which may affect feature extraction. Baraldi *et al.* (2017), Sah *et al.* (2020) explore such an issue by proposing boundary detection mechanisms to consider different segments during description generation.

Despite the impressive performance obtained by these methods, they most often suffer from the hallucination of actions or objects, i.e., the model mentions objects that are not in the

input video. Hallucinations in video captioning approaches may be caused by inadequate visual features extracted from pre-trained models, improper influences of source and target contexts during multi-modal fusion, or exposure bias in the training strategy Ullah e Mohanta (2022).

To overcome the hallucination problem, Zhou *et al.* (2019) proposed a grounding-based video description approach integrating a region detection pipeline to better explain the video descriptions. The authors also introduce the ActivityNet Entities dataset, which provides entity-level bounding box annotations on the ActivityNet Captions dataset. Nevertheless, such alignments were inaccurate in representing the interactions and relationships amongst objects and actions presented in videos.

Unlike Zhou *et al.* (2019), which focused on object hallucinations, Lu e Gao (2022) proposed a scene-graph guidance method to better overcome hallucination problems by considering objects and actions. The method combines video features and scene-graph representation with a two-layer LSTM structure with attention mechanisms to generate video descriptions.

The use of Graph Neural Network (GNN) for representing a video scene as a graph and exploring the relationship between entities and region proposals was also investigated (ZHANG; PENG, 2020; PAN *et al.*, 2020; ZHANG *et al.*, 2020a; LU; GAO, 2022; LI *et al.*, 2022b).

In Zhang e Peng (2020), the authors proposed the object-aware spatio-temporal graph approach. The encoder module consists of three sub-modules to capture temporal relation, spatial relation, and the global context. The Temporal Relation component captures the temporal trajectory of each detected object through the video frames using a bidirectional temporal graph. The Spatial Relation component employs Graph Convolutional Network (GCN) to process spatial relations and obtain meaningful features that encode intra-frame interactive information. The global context is extracted using a Convolutional GRU. The decoder module consists of two GRU with attention mechanisms for temporal and spatial relation information, respectively. It utilizes Vector of Locally Aggregated Descriptors (VLAD) representations of objects and frames, and another GRU is employed to generate the description.

Zhang *et al.* (2020a) also computed the relationship between objects by representing the video as a graph. In their approach, each object detected in a video frame was modeled as a node, and the relationship between objects was learned during the training process using the GCN model. The description generation consisted of an LSTM with attention mechanisms under the guidance of the Teacher-Recommended Learning (TRL), which integrates External Language Model (ELM) during the decoder step. The models using graph model representation

have presented promising results by recognizing more detailed objects and their relationships.

In Krishna *et al.* (2017), it was introduced the Video Dense Captioning dataset to the new task called dense-captioning events. Unlike the video captioning task, which contains only one main activity to be described, dense-video captioning aims to detect many actions in a video and generate a sentence for each video segment. The ActivityNet dataset covers a wide range of complex human activities and averages 3.65 events per video. Their proposed approach involves detecting and identifying all events in a given video and describing them in natural language. Their proposed approach uses Deep Action Proposals (DAP) (ESCORCIA *et al.*, 2016) to localize temporal event proposals and a caption module based on LSTM to generate a sentence for each event proposal.

Building from the idea of video-dense captioning, Li *et al.* (2018) proposed a unified approach to detect temporal events and then generate a sentence for each detected event by jointly training them in an end-to-end manner. The Temporal Event Proposal (TEP) integrates three components (event classification, temporal coordinate regression, and descriptiveness regression) to produce a set of event proposals with scores for the presence of an event and descriptiveness in the proposals. The candidate proposals were then ranked, and the Sentence Generator module, based on LSTM, was trained using reinforcement learning techniques to enhance captioning.

To tackle the temporal dependency between events in a video, Mun *et al.* (2019) proposed a framework that explicitly models temporal dependency across events in a video and leverages visual and linguistic context from prior events for coherent storytelling. The event proposal network is based on Single-Stream Temporal (SST) action proposals (BUCH *et al.*, 2017), which provides a representation of each predicted proposal. Then, the proposed Event Sequence Generation Network (ESGN) selects a series of correlated events and adaptively determines the number and order of events. Finally, the Sequential Captioning Network (SCN) employs a hierarchical RNN to generate the captions. This suggests that the order of proposal events is essential to generate coherent and accurate descriptions.

Zhang *et al.* (2020b) proposed a Graph-based Partition-and-Summarization (GPaS) framework, which generates sentences from short video segments and summarizes them in one sentence. The Bidirection Single-Stream Temporal (Bi-SST) model proposed by Wang *et al.* (2018b) was used to predict temporal proposals for possible actions. Each predicted proposal is partitioned into small video segments to extract visual and textual features for a satisfactory description. These descriptions are then summarized using both visual and textual features. The

summarization module uses GCN and LSTM to explore the relationship of words across semantic levels and generate a single sentence for the proposal event. This way, the summarization explores different levels of semantic meanings and can help generate better sentences.

A deep learning model architecture called Transformer, initially proposed by Vaswani *et al.* (2017), has been employed to tackle various NLP problems and was also adapted to the video captioning task. The transformer architecture consists of an encoder-decoder structure. The encoder takes an input sequence and processes it by applying self-attention mechanisms, allowing the model to weigh the importance of different words or tokens based on their context within the sequence. On the other hand, the decoder generates the output sequence by attending to the relevant parts of the input sequence. The key component of the transformer architecture is the self-attention mechanism, also known as scaled dot-product attention. It computes attention scores between all pairs of words in a sequence and uses these scores to weigh the contributions of each word during the encoding or decoding process. This attention mechanism enables the model to focus on relevant information and improve its ability to understand and generate coherent sequences.

Motivated by the high computational cost faced by RNNs and their difficult of capturing long term dependencies, (ZHOU *et al.*, 2018b) proposed a Transformer-based approach for the dense video captioning task. They proposed a unified end-to-end approach, composed of three components: a video encoder, a proposal decoder, and a captioning decoder. The video encoder is composed of multiple self-attention layers, a type of attention mechanism that allows the inputs to interact with each other and find out to whom they should pay more attention. The Temporal Action Proposal (TAP) module, based on ProcNets (ZHOU *et al.*, 2018a), was designed to detect actions in long videos. The captioning decoder module uses Transformers (VASWANI *et al.*, 2017) to generate a sentence for each event proposal.

Chen *et al.* (2018) proposed a Two-View Transformer (TVT) model for video captioning to learn sequential data based on Transformer instead RNN. In particular, fusion blocks are designed and Transformers exploit information from motion, audio, and spatial features with attention mechanisms. Experimental results showed that the Transformer with the fusion features proposed achieved competitive performance with the state-of-the-art approaches.

Since Transformer has been used in video captioning and has also achieved state-of-the-art results on Neural Machine Translation (NMT) task(MARUF *et al.*, 2021), many researchers have recently studied the use of transformers in the video captioning task (LIN *et al.*, 2022;

MAN *et al.*, 2022; LI *et al.*, 2022; YANG *et al.*, 2021a). Despite the promising and competitive results achieved by Transformer, many papers published in recent years still employ LSTM to generate captions, as can be seen in Appendix A.

The approaches presented so far hold the closed-set assumption that all possible events are known during the train or test phase. However, new human actions may arise over time in real-world dynamic environments.

Wang *et al.* (2019b) introduced the zero-shot video captioning task, which aims to describe out-of-domain videos of unseen activities. The proposed Topic-Aware Mixture of Experts (TAMoE) explores external sources, e.g., Wikipedia and WikiHow, to transfer the knowledge learned from seen topics to unseen topics and thus. Although such an approach may describe unseen videos during training, they fail to describe videos unrelated to the known topics used in training. Moreover, the vocabulary used to describe videos is fixed and may not properly describe such videos with unknown actions.

An open world scenario, where actions and objects are created at will, requires an approach to recognize known classes seen during the training stage accurately and deal with unknown classes not seen during the training or validation stage and without specific vocabulary in the corpus. In this sense, a video description approach is desired to describe known events correctly and incrementally learn unknown events. This thesis proposes a method for video captioning approach in an open-world scenario. To the best of our knowledge, there have been no studies for the video description task in an open world setting up to this point. Table 1 presents an overview of video description methods described in this section.

Table 1 – Summary of video captioning studies. S denotes Spatial features, T denotes Temporal features, R denotes visual Relations, MM denotes Multimodal, NV denotes Novel Actions, EDL denotes Event Detection and Localization, OW denotes Open-World, and LM denotes Language Model.

N.	Author/Year	S	T	R	MM	NV	EDL	OW	LM	Dataset
1	Venugopalan *et al.* (2015b)	AlexNet variant							LSTM	MSVD
2	Venugopalan *et al.* (2015a)	VGG-16	optical flow						LSTM	MSVD, MPII-MD, MVAD
3	Jin *et al.* (2016)	VGG-16	C3D		*				LSTM	MSR-VTT
4	Baraldi *et al.* (2017)	ResNet-50	C3D, LSTM						GRU	M-VAD, MPII-MD e MSVD
5	Krishna *et al.* (2017)		C3D				*		LSTM	ActivityNet Captions
6	Song *et al.* (2017)	ResNet-152							LSTM	MSR-VTT, MSVD
7	Tu *et al.* (2017)	Faster R-CNN, GoogLeNet	C3D						LSTM	MSR-VTT, MSVD
8	Chen *et al.* (2018)	ResNet-152, NasNet	I3D		*				Transformer	MSR-VTT, MSVD
9	Li *et al.* (2018)		C3D		*		*	LSTM	ActivityNet Captions	
10	Xiao e Shi (2018)	Inception-V3	LSTM		*				LSTM	MSVD
11	Wang *et al.* (2018b)		C3D				*		LSTM	ActivityNet Captions
12	Zhou *et al.* (2018b)	ResNet-200	optical flow				*		Transformer	ActivityNet Captions, YouCook2
13	Aafaq *et al.* (2019)	IRv2	C3D		*				GRU	MSR-VTT, MSVD
14	Du *et al.* (2019)	ResNet-152	C3D, optical flow		*				LSTM	MSR-VTT, MSVD
15	Francis e Huet (2019)	ResNet-152	I3D						LSTM	MSR-VTT, MSVD
16	Mun *et al.* (2019)		C3D, GRU				*		LSTM	ActivityNet Captions
17	Pei *et al.* (2019)	ResNet-101	ResNeXt-101						GRU	MSR-VTT, MSVD
18	Wang *et al.* (2019b)		I3D, LSTM		*	*			LSTM	ActivitiNet Captions, MSR-VTT
19	Zhou *et al.* (2019)	Faster R-CNN, ResNeXt-101, ResNet-101	LSTM						LSTM	ActivityNet-Entities
20	Gao *et al.* (2020a)	Resnet-152	C3D						LSTM	MSVD/MSR-VTT/LSMDC
21	Pan *et al.* (2020)	Faster R-CNN, ResNet-101	I3D	STG					Transformer	MSR-VTT, MSVD
22	Sah *et al.* (2020)	ResNet-152	optical flow		*				LSTM	M-VAD, MSR-VTT, MSVD
23	Yan *et al.* (2020)	GoogLeNet, Resnet-152	C3D		*				LSTM	MSR-VTT, MSVD
24	Zhang e Peng (2020)	ResNet-200	VLAD, ConvGRU	GCN					GRU	MSR-VTT, MSVD
25	Zhang *et al.* (2020a)	Faster R-CNN (features), IRv2	C3D	GCN					LSTM	MSVD/MSR-VTT/Vatex
26	Zhang *et al.* (2020b)		C3D, LSTM				*		LSTM	ActivityNet Captions

N.	Author/Year	S	T	R	MM	NV	EDL	OW	LM	Dataset
27	Ahmed *et al.* (2021)	VGG-16, InceptionV3, Xception, faster R-CNN	I3D						LSTM	MSR-VTT, MSVD
28	Perez-Martin *et al.* (2021)	ResNet-152	ECO, d R(2+1)D		*				LSTM	MSR-VTT, MSVD
29	Ryu *et al.* (2021)	ResNet-101	ResNext-101,						LSTM	MSR-VTT, MSVD
30	Tu *et al.* (2021)	VGG, ResNet-152			*				LSTM	MSVD, MSR-VTT
31	Yang *et al.* (2021a)	ResNet-101	ResNeXt-101						Transformer	MSR-VTT, MSVD
32	Lin *et al.* (2022)		I3D		*				Transformer	MSVD, MSR-VTT, VATEX, TVC, YouCook2
33	Man *et al.* (2022)	ResNet-200, BN-Inception	optical flow		*				Transformer	ActivityNet, YouCook2, and VideoStory
34	Ghaderi *et al.* (2022)		Swim video transform		*				Transformer	MSR-VTT, MSVD, Vatex
35	Ji *et al.* (2022)	Inception-V4							LSTM	MSR-VTT, MSVD
36	Li *et al.* (2022b)	IRv2, ResNet-152	optical flow, I3D	GCN	*				LSTM	Charades, MSR-VTT, MSVD
37	Li *et al.* (2022)	IRv2	I3D	LSTG					Transformer	MSR-VTT, MSVD
38	Lu e Gao (2022)	ResNet-152, ResNet-200		GCN	*				LSTM	ActivityNet Captions, Charades
39	Ullah e Mohanta (2022)	ViTL, Faster-RCNN	C3D						LSTM	MSR-VTT, MVSD
40	**Ours**	**ResNet-101, InceptionResnet-v2**	**ResNeXt-101, I3D**			*		*	**LSTM**	**MSR-VTT-subet, Liris, ActivityNet Captions**

Source: Developed by the author.

2.8.2 Datasets for Video Captioning

An important aspect of machine learning methods is the existence of labeled datasets used for training purposes. In the context of the video captioning task, a dataset with a massive number of videos with one or more different descriptions per video is desired. This section presents the datasets most used or cited in the studies found during the literature review.

ActivityNet Captions (KRISHNA *et al.*, 2017) dataset contains 20,000 videos taken from the ActivityNet dataset (HEILBRON *et al.*, 2015), in which each video has, on average, 3.65 temporally localized sentences and a total of 100k sentences. It was proposed to the dense video captioning task, which aims to generate multiple informative and diverse sentences for a video containing short, long or even overlapping events.

Charades (SIGURDSSON *et al.*, 2016) dataset provides 27,847 descriptions of 9,848 videos. Each video has an average length of 30 seconds in 15 types of indoor scenes. There are also available 66,500 temporally localized intervals for 157 action classes and 41,104 labels for 46 object classes. It was proposed for activity understanding, including action classification, localization, and video descriptions.

Large Scale Movie Description Challenge (LSMDC) (ROHRBACH *et al.*, 2017) dataset is based on MPII Movie Description (MPII-MD) and Montreal Video Annotation Dataset (M-VAD). There are two versions of the dataset: LSMDC 15, which contains 118,114 sentences of 118,081 clips from 200 movies, and was made available in a challenge held in conjunction with the International Conference on Computer Vision (ICCV) in 2015; and the LSMDC 16, which contains 128,118 sentences of 128,085 clips from 200 movies, and was made available in a challenge held with the European Conference on Computer Vision (ECCV) in 2016. The dataset is available for download with restricted access, for scientific or research use only, by signing the Agreement term available on the challenge website.

M-VAD (TORABI *et al.*, 2015) contains 55,904 sentences of 48,986 clips from 92 video movies. The authors provide an official training/validation/test split, consisting of 38,949, 4,888, and 5,149, respectively. The dataset is available for download with restricted access, for scientific or research use only, by signing the Agreement term available on the challenge website.

MPII-MD (ROHRBACH *et al.*, 2015) contains 68,375 sentences of 68,337 clips from 94 video movies. The dataset is also available for download with restricted access, for scientific or research use only, by signing the Agreement term available on the challenge website.

MSR-VTT (XU *et al.*, 2016) contains 200,000 sentences of 10,000 clips from 7,180 videos. With an average of 20 different sentences per clip, it is publicly available on the internet, and it is the second most used dataset available. The authors suggest splitting the original dataset into three independent subsets: 6,513 videos for training, 497 videos for validation and 2,990 videos for testing.

Microsoft Research Video Description Corpus (MSVD) (CHEN; DOLAN, 2011) contains 70,028 sentences of 1,970 clips. It is also publicly available on the internet, and it is the most used dataset for training and evaluating video captioning methods. The commonly training/validation/test split firstly proposed by Guadarrama *et al.* (2013) consists of 1,200, 100, and 670, respectively.

Vatex (WANG *et al.*, 2019a) contains 412,690 sentences, in both English and Chinese languages, of 41,269 clips taken from the Kinetics-600 dataset (KAY *et al.*, 2017). The authors provide an official training/validation/test split, consisting of 25,991, 3,000, and 6,000, respectively. Also, a secret test set with 6,278 human-annotated captions was held out for challenge purposes.

TACos-Multilevel (ROHRBACH *et al.*, 2014) contains 52,593 descriptions of 14,105 video clips about person's cooking procedures. It provides three levels of detailed descriptions for complex videos: one sentence for a coplex event; short sentence for a video segment; and detailed description for each step of the cooking procedures.

YouCook2 (ZHOU *et al.*, 2018a) contains 15,400 sentences of video clips in 2000 long untrimmed videos downloaded from YouTube which are all instructional cooking recipe videos; It is the largest task-oriented, instructional video dataset in the vision community.

Although the UET-Surveillance (DILAWARI; KHAN, 2019), Sports Video Narrative Dataset (YAN *et al.*, 2022), SVCDV (QI *et al.*, 2018), and TrecVid 2016 (SALEEM *et al.*, 2019), TVC(LEI *et al.*, 2020) , VideoStory(GELLA *et al.*, 2018), SVN(YAN *et al.*, 2022), and EmVidCap(WANG *et al.*, 2021) datasets were used by some approaches found in the literature, as shown in Table 1, they were not considered for this research because they currently are not available or were proposed for specific tasks different than video captioning task.

LSMDC is the largest dataset in terms of number of videos (118,081) with one sentence for each video. On the other hand, the ActivityNet Captions has a much larger vocabulary (50 times more different words in the corpus), as summarized in Table 2. The vocabulary size usually indicates a higher diversity of objects, actions and information details and, therefore,

can be used to describe, with more diversity, the video scenes. However, they usually are sparse and unbalanced and therefore, may produce more noise in the alignments. A small vocabulary tends to have similar syntactic structure and can also lead to diverse descriptions using different combinations of words.

Table 2 – Datasets used for evaluating video description approaches.

Dataset	Domain	N. Videos clips	N. Sentences	Vocabulary
ActivityNet Captions	Social Media	20,000	100,000	1,348,000
Charades	Daily Activities	9,848	27,847	4,144
LSMDC	Movies	118.081	118,114	25,610
M-VAD	Movies	48,986	55,904	17,609
MPII-MD	Movies	68,337	68,375	24,549
MSR-VTT	Open	10,000	200,000	29,316
MSVD	Open	1,970	70,028	13,010
TACoS-MultiLevel	Cooking	14,105	52,593	2,000
YouCook2	Cooking	2,000	15,400	2,600
Vatex	Open	41,269	412,690	35,609

Source: Developed by the author.

Although large datasets are available for the video description task, the most used datasets are MSVD and MST-VTT. They are challenging enough by containing descriptions of open-domain videos and have a small vocabulary, achieving more easily satisfactory results.

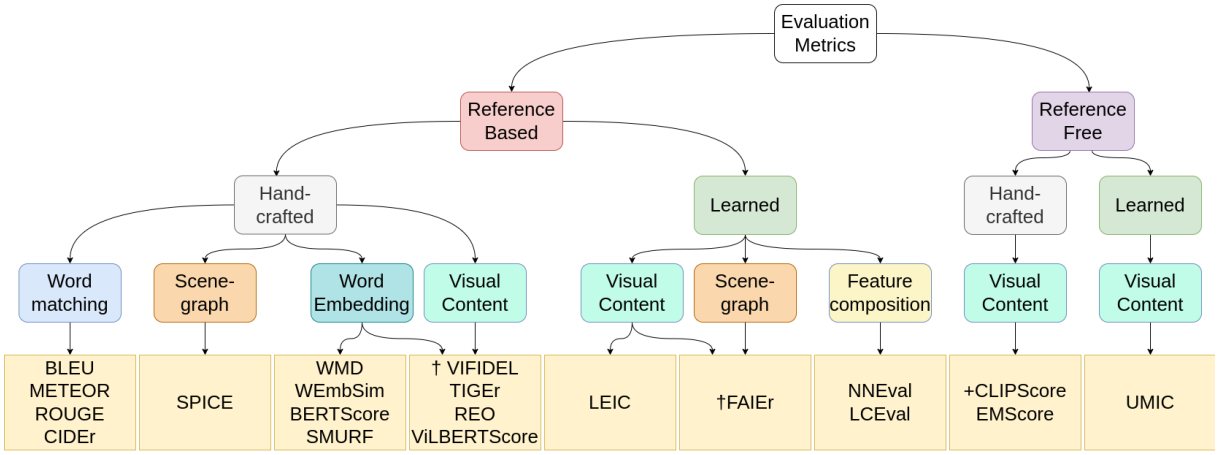
ActivityNet Captions contains untrimmed videos from YouTube with annotated temporal sentences. Each sentence covers one segment of the video, describing various occurring events. These events can occur over long or short periods of time. It also has an overlap of 10% of the temporal descriptions, indicating the presence of concurrent events. The main dataset limitation is that only C3D features for visual frames are provided. Although there are downloadable video URLs, many of them are unavailable. Moreover, the testing set labels are not publicly available, and the performance is usually reported on the validation set.

Similar to ActivityNet Captions, the YouCook2 dataset was also created from Youtube Videos and has the same limitation regarding video availability. The authors only make available preprocessed frame features extracted by the ResNet network. Also, the descriptions of 210 testing videos are unavailable, and the validation set is often used to report the performance.

2.8.3 Metrics for video captioning evaluation

A video shot can be described in different ways and with different levels of detail. Therefore, automatically evaluating a generated sentence is a challenging task.

Figure 11 – Taxonomy of evaluation metrics.



+ Metric may take into account both visual content and reference sentences.
† Metric may take into account only the visual content.

Source: Developed by the author.

This section analyzes the automatic evaluation metrics commonly employed in the video caption task. In addition, some metrics are also presented that were explicitly proposed for image captioning, but that can also be useful and promising for the video captioning task. The taxonomy and summary of evaluation metrics presented in this Section were published in (INÁCIO, 2023).

The analyzed metrics can be divided into two primary categories: reference-based and reference-free. Then, each category can additionally be split into learned and hand-crafted subcategories. The hand-crafted approaches employ deterministic similarity measures between a candidate and the reference sentences or the input visual content (video or image). The learned methods usually require training a (neural network) model to predict the likelihood of a candidate caption being a machine or human-generated description.

In the proposed taxonomy, we also consider the way these metrics encode the sentences to compute the similarity score, which can be divided into four main ways:

1. Word-matching: when n-grams are compared;
2. Scene-graph: when sentences are encoded as a scene-graph prior to comparison;
3. Word embedding: when using a pre-trained word-embedding to encode sentences;
4. Feature composition: when different features are considered.

Moreover, some metrics also include visual content (concepts captured in images) to measure the similarity. They were also categorized in the proposed taxonomy.

2.8.3.1 Reference-based metrics

Existing datasets for video captioning consist of a set of videos paired with captions in natural language, usually written by humans, which describe their visual content. Most metrics used for evaluating video captioning approaches are based on those sentence references. Thus, given a candidate sentence generated by the approach, the metrics evaluate the sentence by measuring its similarity against a set of reference sentences associated with a given visual content. A brief description of the reference-based metrics is presented below.

BLEU (PAPINENI *et al.*, 2002) is a metric proposed for automatic translation to measure the proximity of the reference generated with one or more reference human description. It is based on modified n-grams precision and it is usually computed for n-grams of size 1 to 4 and it is reported as a percentage value. The grammatical correctness or intelligibility is not considered. BLEU is calculated as follows:

$$BP = \begin{cases} 1, & \text{if } c > r \\ \exp^{1-\frac{r}{c}}, & \text{if } c <= r \end{cases} \tag{15}$$

$$BLEU = BP * \exp(\sum_{n=1}^{N} \frac{\log(p_n)}{N}) \tag{16}$$

where BP is a brevity penalty to penalize candidate sentences longer than their references, c is the length of a candidate sentence, r is the length of a reference sentence, N is the length of n-grams, and p_n is the geometric average of the modified n-grams. A high score in this metric may be associated with a large number of references. According to Denkowski e Lavie (2010 apud SELJAN *et al.*, 2012), BLEU scores above 0.30 generally reflect an understandable sentence, and above 0.50 reflect good and fluent candidate sentences.

METEOR (LAVIE; AGARWAL, 2007) is also a metric initially proposed for automatic translation and designed to address the weakness observed in the BLEU metric, including the lack of recall, the use of higher-order n-grams, the lack of explicit word-matching between candidate and reference sentences, and the use of geometric averaging of n-grams. It consists of creating alignment between unigrams from candidate and reference sentence. Each unigram from the candidate can have zero or one mapping to a unigram from the reference sentence. The metric is based on the precision, recall, and harmonic mean and consists of creating alignment between unigrams from candidate and reference sentence. The word matching supports morphological

variants including stemming, and synonyms. Once the alignment is computed, the score is computed as follows:

$$Penalty = 0.5 * (\frac{c}{u_m}) \tag{17}$$

$$F_{mean} = \frac{10PR}{R + 9P} \tag{18}$$

$$METEOR = F_{mean} * (1 - Penalty) \tag{19}$$

where c is the number of adjacent unigrams between candidate and reference sentence, u_m is the number of unigrams mapped, P is the precision (ratio between the number of unigrams from the candidate found in reference over the number of unigram of the candidate sentence), R is the recall (ratio between the number of unigrams from the candidate found in reference over the number of unigrams of the reference sentence). METEOR score provides a better correlation with human judgments than the BLEU score.

CIDEr (VEDANTAM *et al.*, 2015) is a consensus-based metric originally proposed for image captioning and measures the similarity of a generated sentence against a majority of a set of ground truth sentences written by humans. It employs morphological variations by changing each word in its stem or root form to resolve word-level correspondences. Then, the Term Frequency-Inverse Document Frequency (TF-IDF) is performed for each n-gram. Finally, the average cosine similarity is computed between the candidate and the reference sentences, as follows.

$$CIDEr_n(c_i,S_i) = \frac{1}{m} * \sum_{i=1} g^n(c_i).\frac{g^n(s_{ij})}{||g^n(c_i)||.||g^n(s_{ij}||} \tag{20}$$

$$CIDEr(c_i,S_i) = \sum_{n=1}^{N} \frac{CIDEr_n(c_i,S_i)}{N} \tag{21}$$

where $g^n(c_i)$ and $g^n(s_{ij})$ corresponds to all *n*-grams of length n from candidate and reference sentences, respectively, and $||g^n(c_i)||$ and $||g^n(s_{ij})||$ are the magnitude of the vectors $g^n(c_i)$ and $g^n(s_{ij})$, respectively.

The CIDEr-D is a variation of CIDEr commonly used to evaluate video description approaches. In this metric, the stemming step was removed, a Gaussian penalty based on the difference between candidate and reference sentence lengths was introduced, and added clipping

to the n-gram counts in the CIDEr numerator. These modifications aims to avoid sentences with high scores but with poor results when judged by humans.

ROUGE (LIN, 2004) is a metric proposed initially to determine the quality of summarization. It consists of four different variations: ROUGE-N (N-gram Co-Occurrence Statistics), ROUGE-L (Longest Common Subsequence A), ROUGE-W (Weighted Longest Common Subsequence), and ROUGE-S (Skip-Bigram Co-Occurrence Statistics). ROUGE-L is used for Video Descriptions, and it is applied in two different levels: sentence-level and summary-level. It computes the recall and precision scores of the Longest Common Subsequences (LCS) between a sentence X of length m and a sentence Y of length n. The similarity between two sentences is given as follows:

$$R_{lcs} = \frac{LCS(X,Y)}{m} \tag{22}$$

$$P_{lcs} = \frac{LCS(X,Y)}{n} \tag{23}$$

$$F_{lcs} = \frac{1 + \beta^2 R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \tag{24}$$

where LCS(X, Y) is the length of the longest common subsequence between X and Y, and β is a weighting factor that specifies different importance to precision and recall and is usually set to a value of 1.2 (CHEN *et al.*, 2015).

Semantic Propositional Image Caption Evaluation (SPICE) (ANDERSON *et al.*, 2016) is the most recent metric proposed for image captioning. It was designed to tackle the limitations of the existing automatic evaluation metrics based on n-grams, such as BLEU, METEOR, and CIDEr. These metrics usually assigns a low score to a generated sentence that conveys almost the same meaning of the reference but has no words in common.

The caption quality is computed based on the F1-score using a graph-based semantic representation called scene graph, which uses a dependency parse tree to encode objects, their attributes, and relationships. These encoded concepts are organized as tuples containing one, two, or three elements. The similarity between tuples in two scene graph is computed as follows:

$$SPICE(c,S) = \frac{2.P(c,S).R(c,S)}{P(c,S) + R(c,S)} \tag{25}$$

where $P(c,S)$ is the precision of the matching tuples, and $R(c, S)$ is the recall of the matching tuples. During the match analysis between tuples, synonym, and lemmatization techniques are considered to allow the match of words with different inflection forms.

BLEU, METEOR, ROUGE, and SPICE range from 0 to 1, with 1 as identical to the references. CIDEr ranges from 0 to 10, with 10 as identical to the reference.

Word Mover's Distance (WMD) is a distance measure proposed by Kilickaya *et al.* (2017) to calculate the dissimilarity between two text documents. It was inspired by the "Earth Mover's Distance", employing a solver of the "transportation problem".

This metric aimed to assess the semantic distance between documents by representing the words as word embedding vectors. It calculates the minimum distance that words in one document should travel to the words in another document.

This metric was not designed for image or video captioning evaluation. However, it has been used to evaluate image captioning approaches (LAINA *et al.*, 2019) and, over time, it has inspired the development of other metrics.

WEmbSim. Similar to WMD, WEmbSim (SHARIF *et al.*, 2020) uses word embeddings to encode the words in an embedding space. Then each sentence is represented by a feature vector computed using the Mean of Word Embeddings (MOWE). Then, the distance between two sentences is computed by the cosine distance. This metric was designed to automatically evaluate image captioning systems in terms of system-level performance. Similar to SPICE, WEmbSim neglects fluency and focuses only on semantics and may fail to distinguish sentences with the same words in different ordering.

Bidirectional Encoder Representations from Transformers Score (BERTScore) (Z. *et al.*, 2020) is an automatic metric for machine translation and image captioning systems. Tokens are represented by contextual embeddings using the Bidirectional Encoder Representations from Transformers (BERT) model (DEVLIN *et al.*, 2019), which can generate different vector representations for the same word in different sentences. Then, the pairwise cosine similarity is computed, and a greedy matching procedure is used to maximize the matching similarity score.

SeMantic and linguistic UndeRstanding Fusion (SMURF) (FEINGLASS; YANG, 2021) is an automatic evaluation metric that combines a novel semantic evaluation algorithm Semantic Proposal Alikeness Rating using Concept Similarity (SPARCS) and novel fluency evaluation algorithms Stochastic Process Understanding Rating using Typical Sets (SPURTS) and Model-Integrated Meta-Analysis (MIMA) for both caption-level and system-level analysis. A

Transformer-based model such as BERT or Robustly optimized BERT approach (RoBERTa) (L. *et al.*, 2019) is used to extract features from texts and capture both the syntax and morphology of the text.

VIsual Fidelity for Image Description EvaLuation (VIFIDEL) (MADHYASTHA *et al.*, 2019) is inspired by the WMD metric and estimates the faithfulness of a generated caption concerning the content of a given image. It measures the similarity between label objects detected in the image and the words in the generated descriptions using WMD. It also can be used with reference descriptions when available.

Text-to-Image Grounding based metric for image caption Evaluation (TIGEr) (JIANG *et al.*, 2020) is a metric proposed for evaluating image captioning systems that consider both image content and sentence references.

The metric uses a pre-trained image-text grounding model to compute features from an image-sentence pair in a common semantic space, indicating the similarity between image and sentence. The final score is computed by combining two metric systems: Region Rank Similarity (RRS) and Weight Distribution Similarity (WDS).

Relevance, Extraness, Omission (REO). Unlike other metrics that compute a numeric score that indicates the quality of a candidate sentence or video captioning system, REO provides an informative assessment. This measure was introduced by Jiang *et al.* (2019) and evaluates the quality of captions, generating scores from three different perspectives: Relevance, Extraness, and Omission. Similar to the TIGEr metric, REO first extracts features from images and sentences (reference and candidate) using a pre-trained model to build a multimodal semantic space. The relevance score is computed by the cosine similarity distance between the candidate and reference features. Extraness and Omission scores are calculated by performing an orthogonal projection of image features and reference sentence features. A final score can be computed by averaging the three aforementioned scores.

Vision-and-Language BERT Score (ViLBERTScore) (LEE *et al.*, 2020) was inspired by the excellent performance of word-embedding techniques, especially the BERTScore model, in many text-generation tasks. It computes image-conditioned embeddings for each token using Vision-and-Language BERT (ViLBERT) (LU *et al.*, 2019a) from both generated and reference texts. A cosine similarity among the pair of tokens from the candidate and reference caption is computed. The greedy matching process between these tokens is expressed via the cosine similarity of their embeddings. The best matching token pairs are used for computing precision,

recall, and F1-score.

Learning to Evaluate Image Captioning (LEIC) (CUI *et al.*, 2018) is a learning-based discriminative evaluation metric trained to distinguish between human and machine-generated captions. The predicted and reference captions (when available) and images are encoded as feature vectors and then fed into a softmax classifier, which outputs the probability of being a human-written or machine-generated description.

Fidelity and Adequacy ensured Image caption Evaluation metric (FAIEr) (WANG *et al.*, 2021) is another learning-based metric to evaluate the fidelity and adequacy of captions generated by image caption systems. It employs the same scene graph parser used by SPICE to represent sentences as a textual scene graph. To build a visual scene graph, an object detector is used to detect and extract features of objects from an image. Each object detected is a graph node, and the relationship-level representation is encoded using a GCN. Visual and reference scene graphs are fused together by using an attention mechanism. The final score is computed by measuring the similarity between two scene graphs at the object and relationship levels.

Neural Network based Evaluation Metric (NNEval) (SHARIF *et al.*, 2018) is also a learning-based metric proposed to evaluate image captioning systems. This metric considers lexical and semantic information using a composition of well-established metrics such as BLEU, METEOR, CIDEr, SPICE, and WMD. Instead of directly using candidate and reference sentences to train the metric, the set of composed features of the scores generated by each individual metric is considered. Then, the feature vector is used to feed a feed-forward neural network, which outputs the probability of an input sentence being human-generated.

Learned Composite Metric for Caption Evaluation (LCEval) (SHARIF *et al.*, 2019) is an extension of the NNEval metric. It is also a learning-based metric made up of different computed metrics. However, unlike NNEval, which combines all features into a feature vector, LCEval creates three feature subgroups based on the lexical, semantic, and syntactic properties. The lexical features include BLEU, METEOR, ROUGE-L, and CIDEr. The semantic features consider SPICE, WMD, and MOWE. Finally, the syntactic features are extracted using the Head Word Chain Matches (HWCM), which captures the syntactic similarity between sentences using the tree structure of the sentences.

2.8.3.2 Reference-free metrics

Due to the known limitations of the existing metrics based on reference sentences, mainly regarding the difficulty of obtaining several possible ways of describing the same visual content, some reference-free metrics were recently proposed. In such metrics, visual and textual features are extracted using pre-trained neural network models for the image-text matching task. Then, a similarity score is computed. A brief description of the reference-free metrics studied is described below.

Contrastive Language-Image Pre-training Score (CLIPScore) was introduced by (HESSEL *et al.*, 2021) for assessing image captioning systems without reference sentences. It uses the Contrastive Language-Image Pre-training (CLIP) (RADFORD *et al.*, 2021) model, a cross-modal retrieval model pre-trained on a dataset comprising 400M pairs of images and captions, to extract features from images and candidate sentences. Then, the final score is computed by measuring the cosine similarity between features. Besides, the metric can be extended to incorporate references when available.

Unreferenced Metric for Image Captioning (UMIC) (LEE *et al.*, 2021) is also a free-reference metric proposed to evaluate the quality of sentences generated by image captioning systems. It uses image features extracted from the UNiversal Image-TExt Representation learning (UNINTER) (CHEN *et al.*, 2020), a pre-trained model to predict alignment between images and texts. The model is trained to distinguish between the reference sentences and negative captions using synthetic negative samples.

Embedding Matching-based Score (EMScore) (SHI *et al.*, 2022) is a free-reference metric proposed for evaluating video captioning approaches. It uses the pre-trained image-language model CLIP, mentioned before, to extract video and text embeddings. The final score is computed by combining a fine-grained score using a sum of cosine similarities between frames and word embeddings, and a coarse-grained score using the similarity between the global embeddings of the video and the candidate caption. The reference sentences can also be considered as an extended metric called EMScore_ref.

3 PROPOSED VIDEO DESCRIPTION METHODS

This Chapter presents the methods proposed for performing video descriptions in an open-world scenario. As discussed in Section 1.1, a video description approach in an open-world scenario may be able to describe known events, detect unknown events, and learn unknown new events by updating the existing model without re-training the whole model from scratch.

The first Section of this Chapter presents a method for performing video descriptions in an open-set scenario. The main objective is to describe concurrent known events and detect unknown ones. The method and results of this Section were published in (INÁCIO *et al.*, 2021).

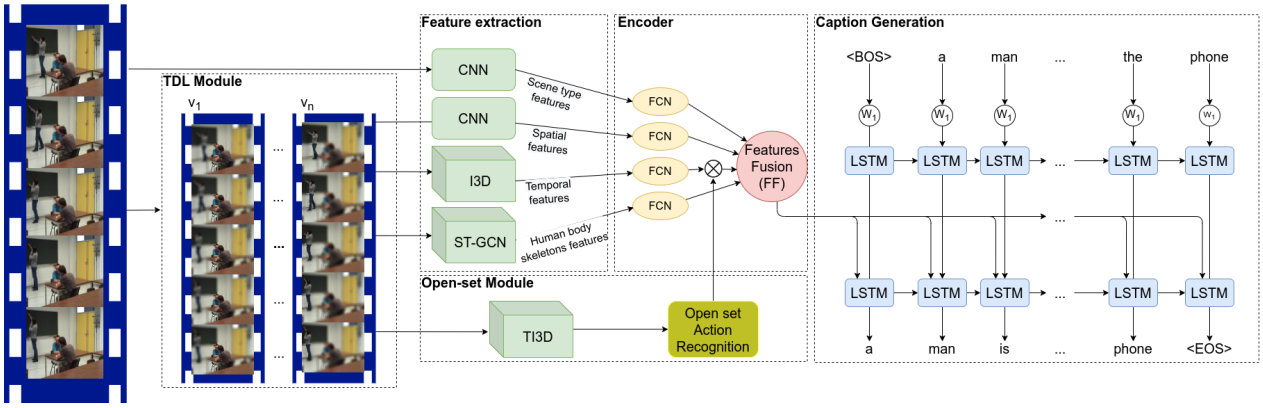
The second Section of this Chapter presents an incremental learning method for the video description approach. The proposed method focuses on updating the existing method to learn how to describe unknown events without forgetting the known ones. Note that it is essential to provide descriptions in natural language for the detected unknown videos before the incremental learning training stage. We assume that this task will be accomplished by human intervention.

3.1 VIDEO DESCRIPTION IN AN OPEN-SET SCENARIO

3.1.1 Introduction

This Section presents a novel open-set video captioning framework that aims to describe, in natural language, both single and concurrent events occurring in a video. The proposed approach is based on the encoder-decoder framework and uses an open-set action recognition model to detect unknown actions, thus avoiding incorrect descriptions and hallucinations. It also uses a detection-and-tracking-object-based mechanism followed by a background blurring method to define the targets and recognize the concurrent actions to be described. Additionally, we employ the TI3D proposed by (GUTOSKI *et al.*, 2021a), which uses deep metric learning, and the EVM (RUDD *et al.*, 2018) as the open-set classifier. The remainder of this Section is organized as follows. Section 3.1.2 presents the proposed method in detail, and Section 3.1.3 presents the evaluation protocol employed in the experiments.

Figure 12 – An overview of the OSVidCap framework.



Source: Developed by the author.

3.1.2 The proposed method

This Section presents a framework proposed for video description in an open-set scenario called OSVidCap. It comprises five main modules: Target Detection and Localization (TDL), Features extraction, Open set module, Encoder, and Caption Generation. The overall architecture of the method is presented in Figure 12 and detailed as follows.

3.1.2.1 Target Detection and Localization (TDL)

Detecting multiple concurrent events in a given video is essential to adequately describe them in natural language. The TDL module consists of a mechanism designed to detect and track significant moving objects in a given video, which are considered the main concepts of the event. The output of this module consists of video segments for each moving object detected with a blurred background.

More specifically, the TDL module detects and tracks humans but is easily adaptable to other moving objects (such as animals and vehicles). We employ the Yolo-v4 (BOCHKOVSKIY *et al.*, 2020) to detect humans and track them using the Deep SORT method (WOJKE *et al.*, 2017). The human-human or human-object interaction is captured when they overlap in consecutive frames. In such cases, the entities are considered a single region of interest in the final video segment.

Finally, inspired by Tsai *et al.* (2020), we use a background blur method to guide the sentence generator module to focus on each region of interest in each video segment while generating the sentences.

3.1.2.2 Feature extraction

When describing human actions, it is important to consider details of the person, place, and action performed (SHIGETO *et al.*, 2020). Thus, the Encoder module comprises four main classes of features extracted from a given input video, as shown in Figure 12. All these features were extracted using off-the-shelf models, pre-trained on large datasets, which proved to be beneficial for video captioning tasks (VENUGOPALAN *et al.*, 2015b), detailed as follows:

- **Scene type features**: A sample of 16 evenly-spaced frames per video was used to extract the max-pooling features from the last convolutional layer using the VGG model pre-trained on the Places365 dataset [1]. The final representation is a 512-dimensional feature vector.

- **Spatial Features**: For extracting spatial features, we follow (YANG *et al.*, 2021a) and used the ResNet-101 model (HE *et al.*, 2016; RYU *et al.*, 2021), pre-trained on the Imagenet dataset. From a sample of 16 equally spaced frames, we extracted a 2048-dimensional semantic feature vector of each frame from the last pooling layer. Then, an average pooling operation was performed, resulting in the final feature vector of dimension 2048.

- **Temporal Features**: Following (YANG *et al.*, 2021a; RYU *et al.*, 2021), The ResNeXt-101 with 3D convolutions (HARA *et al.*, 2018), pre-trained on the Kinetics dataset (KAY *et al.*, 2017), was used to extract a 2048-dimensional semantic feature vector for every 16 frames (with 50% of overlap) and then, followed an average pooling to obtain a final vector with 2048 features.

- **Human body skeleton features**: We used the ST-GCN model (YAN *et al.*, 2018), pre-trained on the Kinetics dataset, to extract meaningful complementary information for the spatial and temporal features. This is a graph-based model for modeling dynamic skeletons extracted with the OpenPose toolbox (CAO *et al.*, 2017). It is aimed at capturing motion information in dynamic skeleton sequences. We performed a global max-pooling operation over all the skeleton sequences to obtain a single 256-dimension feature vector for a given video. Combining skeleton features with spatial and temporal features was intended to improve the performance in action recognition and, consequently, in the descriptions of the videos (LI *et al.*, 2020).

[1] Weights available at https://github.com/GKalliatakis/Keras-VGG16-places365.git

Except for the scene-type features extracted from the original video frames, all other features were computed with the video segment processed by the TDL module. All of these features are employed in the encoder model to compute the final vector representation of the features.

3.1.2.3 Open set module

This module is based on the EVM model, which is described in Section 2.5 and uses features extracted from the TI3D, described in Section 2.6, for detecting unknown events.

Following the training strategy proposed by Gutoski *et al.* (2021a), semi-hard and hard triplets were mined for each training epoch. Semi-hard triplets are defined as triplets in which the distance between the Anchor and Positive is smaller than the distance between the Anchor and Negative videos, but this distance is smaller than the margin parameter, i.e., $\Theta(f(\mathbf{x}^a), f(\mathbf{x}^p)) < \Theta(f(\mathbf{x}^a), f(\mathbf{x}^n)) < \Theta(f(\mathbf{x}^a), f(\mathbf{x}^p)) + \alpha$. Hard triplets are defined as triplets in which the distance between the Anchor and Positive is larger than the distance between the Anchor and Negative, i.e., $\Theta(f(\mathbf{x}^a), f(\mathbf{x}^p)) > \Theta(f(\mathbf{x}^a), f(\mathbf{x}^n))$. This triplet mining strategy ensures that only triplets with a positive loss w.r.t. Eq. 13 are used during training.

The TI3D network was trained for 20 epochs, updating the triplets every epoch using the hard and semi-hard triplets. The learning rate was set to 0.02, the margin parameter to 0.2, and the batch size to 256. The TI3D network was initialized with the weights of the I3D network.

Then, the EVM was trained using features from training videos extracted using the TI3D network. The output of the module supports the caption generation by signalling whether the action belongs to a known or unknown class. For the EVM, the tail size τ was set to 10% of the number of samples in the train set, the cover threshold for model reduction was set to 0.5, and the probability of inclusion (δ) to 0.5. These parameters were empirically set, based on previous experiments on the datasets used in this study.

3.1.2.4 Encoder

This block aims to derive a feature vector representing the essential concepts to predict the next word for describing the ongoing action in the video. All the previous features extracted from the video were mapped into a common high-level abstract space by a FCN with ReLU activation function, as depicted in Figure 12.

Before Feature Fusion (FF) step, we fuse the output processed by the Open Action Recognition component with the processed Temporal Features ($\mathbf{F}_{tp}$) to consider the unknown action information. Notice that the processed Place-type features ($\mathbf{F}_{p}$), Spatial features ($\mathbf{F}_{sp}$), and Human body skeleton features ($\mathbf{F}_{sk}$) were remained to preserve essential information for caption generation, such as information about the place-type and number of people detected in the scene.

The output calculation of the encoder module provided by the **FF** can be formulated as follows:

$$\mathbf{F}_{p} = \Phi(\mathbf{W}_1 * \mathbf{U}_{p} + \mathbf{b}_1), \tag{26}$$

$$\mathbf{F}_{sp} = \Phi(\mathbf{W}_2 * \mathbf{U}_{sp} + \mathbf{b}_2), \tag{27}$$

$$\mathbf{F}_{sk} = \Phi(\mathbf{W}_3 * \mathbf{U}_{sk} + \mathbf{b}_3), \tag{28}$$

$$\mathbf{F}_{tp} = \Phi(\mathbf{W}_4 * \mathbf{U}_{tp} + \mathbf{b}_4) \otimes \mathbf{O}_{uk}, \tag{29}$$

$$\mathbf{FF} = \mathbf{F}_{p} \odot \mathbf{F}_{sp} \odot \mathbf{F}_{sk} \odot \mathbf{F}_{tp}, \tag{30}$$

in which $\mathbf{W}_1,\ \mathbf{W}_2,\ \mathbf{W}_3$, and $\mathbf{W}_4$ are weight matrices; $\mathbf{U}_{p},\ \mathbf{U}_{sp},\ \mathbf{U}_{sk}$, and $\mathbf{U}_{tp}$ are features from the input modules: scene type, spatial, human body skeleton, and temporal, respectively; $\mathbf{b}_1,\ \mathbf{b}_2,\ \mathbf{b}_3$, and $\mathbf{b}_4$ are the bias vectors; Φ denotes the ReLU activation function; $\otimes$ denotes element-wise multiplication operator; $*$ is the convolution; $\odot$ is the concatenation operator; and $\mathbf{O}_{uk}$ denotes the feature vector provided by the TDL module.

3.1.2.5 Caption Generation

This module consists of the sentence generation and uses two LSTM, a variant of RNN, which works better with long-term dependencies. The first LSTM encodes the preceding sequence of words $S = s_0, s_1, \ldots, s_{t-1}$. The second LSTM predicts the next word based on the output of the first LSTM combined with visual features computed by the Encoder module. The LSTM calculation formula used in this work is given by the following equations:

$$\mathbf{h}_t = tanh(\mathbf{C}_t) * \mathbf{o}_t, \tag{31}$$

$$\mathbf{C}_t = \sigma(\mathbf{f}_t * \mathbf{C}_{t-1} + \mathbf{i}_t * \tilde{\boldsymbol{C}}_t), \tag{32}$$

$$\tilde{\boldsymbol{C}}_t = tanh(\mathbf{x}_t\mathbf{U}^g + \mathbf{h}_{t-1}\mathbf{W}^g), \tag{33}$$

in which $\mathbf{U}^g$ and $\mathbf{W}^g$ are weight matrices; $\mathbf{x}_t$ is the input at time t; $\mathbf{h}_{t-1}$ is the previous state; and $\mathbf{f}_t$, $\mathbf{i}_t$, and $\mathbf{o}_t$ are the forget, input and output gates, respectively. The calculations of unit gates are:

$$\mathbf{f}_t = \sigma(\mathbf{x}_t\mathbf{U}^f + \mathbf{h}_{t-1}\mathbf{W}^f + \mathbf{b}_f), \tag{34}$$

$$\mathbf{i}_t = \sigma(\mathbf{x}_t\mathbf{U}^i + \mathbf{h}_{t-1}\mathbf{W}^i + \mathbf{b}_i), \tag{35}$$

$$\mathbf{o}_t = \sigma(\mathbf{x}_t\mathbf{U}^o + \mathbf{h}_{t-1}\mathbf{W}^o + \mathbf{b}_o), \tag{36}$$

in which $\mathbf{U}^f$, $\mathbf{U}^i$, $\mathbf{U}^o$, $\mathbf{W}^f$, $\mathbf{W}^i$, and $\mathbf{W}^o$ are weight matrices, $\mathbf{b}_f$, $\mathbf{b}_i$ and $\mathbf{b}_o$ are bias vectors, and σ denotes the sigmoid activation function.

3.1.3 Evaluation Protocol

The captions generated by the proposed framework were evaluated using automatic metrics frequently used for comparison with the state-of-the-art methods: BLEU (PAPINENI *et al.*, 2002), METEOR (LAVIE; AGARWAL, 2007), ROUGE-L (LIN, 2004), and CIDEr (VEDANTAM *et al.*, 2015). Although these metrics have limitations and new promising metrics have been proposed recently, as presented in Section 2.8.3, such metrics are widely accepted in the literature to report the efficacy of video description approaches. All metrics were computed using the COCO-caption API (CHEN *et al.*, 2015). BLEU is a metric based on *n*-grams precision modified and measures the predicted sentence proximity with one or more reference descriptions. Following most previous works for video captioning (AAFAQ *et al.*, 2019), we used four-grams with the BLEU metric, which is referred to as BLEU-4. METEOR is based on the precision, recall, and harmonic mean and consists of creating an alignment between uni-grams from candidate and reference sentences. The word matching supports morphological variants including stemming and synonyms. CIDEr is a consensus-based metric and measures the similarity of a generated sentence against a majority of a set of ground-truth sentences. It employs morphological variations by changing each word in its stem (or root form) to resolve word-level correspondences. ROUGE-L computes the recall and precision scores using the LCS technique and tends to reward long sentences with high recall. In our experiments, BLEU, METEOR, and ROUGE metrics were normalized to range from 0 to 100, with 100 as identical to the reference sentence. CIDEr ranges from 0 to 1000, with 1000 as identical to the reference.

3.2 VIDEO DESCRIPTION IN A CLASS-INCREMENTAL LEARNING SETTING

3.2.1 Introduction

Incremental learning techniques aim to continuously learn new tasks over time without forgetting previous ones. In such a scenario, the dataset used for training is divided into a series

of tasks, with the learner only having access to the data of a single task during each training session. In contrast to incremental learning tasks that require a task-ID to perform predictions, class-incremental learning must be capable of distinguishing between all classes from all tasks at inference time, as it lacks access to the task-ID (MASANA *et al.*, 2023).

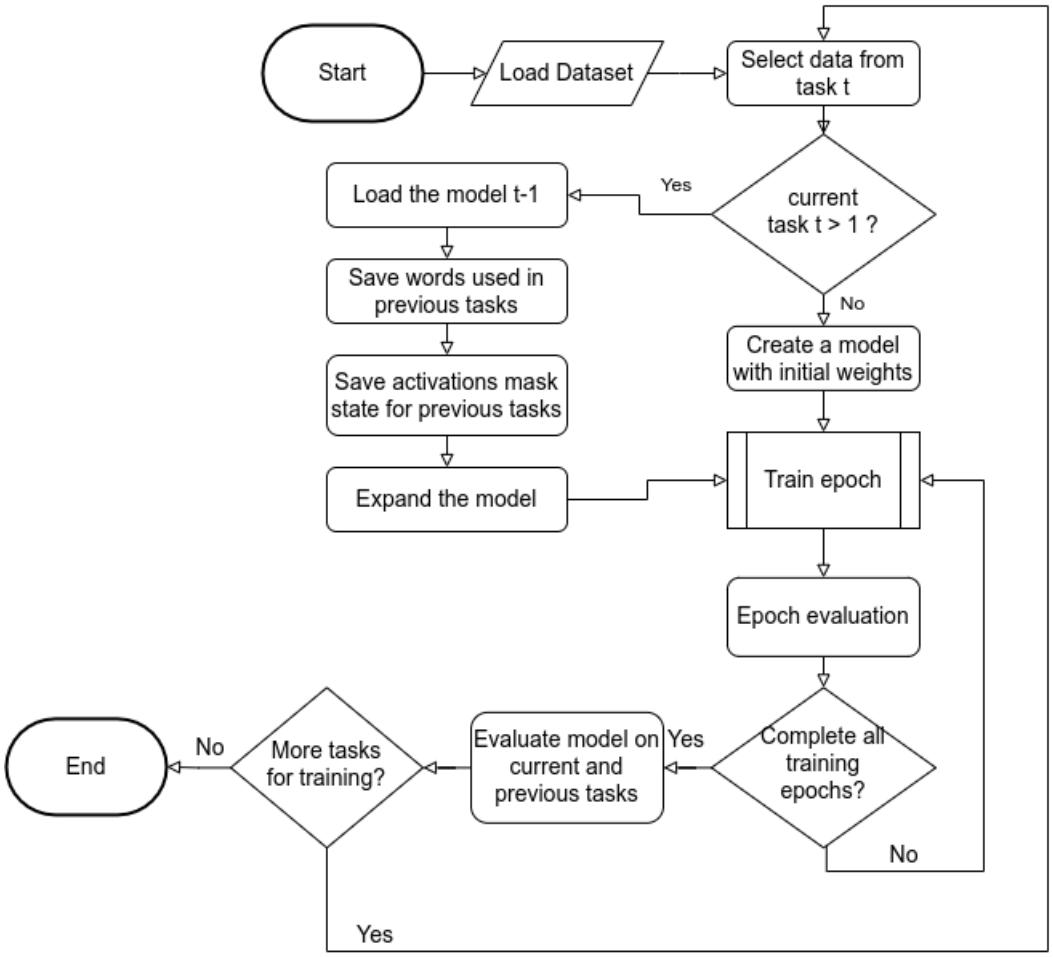
Most class-incremental learning studies focus on the classification problem (BELOUADAH *et al.*, 2021), and little attention has been given to sequential problems. The proposed method presented in this section addresses the problem of video description in class-incremental learning settings. Unlike the classification task, which contains a set of disjoint labels, the video captioning task contains a sequence of words, such as articles, verbs, nouns, and others, with many of these words being shared among tasks. Additionally, synonymous and homonymous words may be present in different tasks, making such tasks more challenging.

The method presented in this section is based on the OSVidCap framework, which uses the LSTM network. To tackle the catastrophic forgetting problem often faced by DL models, attention masks are employed to identify the essential set of neurons in each layer of the caption generator for each task and deactivate them while learning new tasks. Figure 13 presents a high-level overview of the proposed approach.

Firstly, the dataset is split into tasks, with each task containing samples from one or more classes. Here, classes are considered a set of related activities that share characteristics or attributes, allowing grouping and categorizing data instances based on their similarities and assigning them to specific class labels. For instance, classes could include activities such as make-up, dancing, cooking, and talking on the phone. Additionally, each task contains its vocabulary, comprising shared words among tasks and task-specific words exclusively used within that particular task.

Then, the model is sequentially trained from task $t = 1$ to n. When training the first task, the input and output layers are initialized based on the vocabulary size of the current task. Before training a new task ($t > 1$), the network is expanded to accommodate new knowledge from the current task. Additionally, specific neurons are frozen based on their relevance from previous tasks, computed using the attention mask mechanism described in detail in Section 3.2.2. When completing the training for a task t, the performance is reported using the testing data of the current task. Furthermore, it also reported the performance in previous tasks to assess the forgetting rate.

Figure 13 – High-level overview of the proposed class-incremental learning approach.



Source: Developed by the author.

3.2.2 Model Architecture

The proposed method adopts the same encoder-decoder architecture used in the OS-VidCap framework described in Section 3.1.2. The encoder step extracts video features using four off-the-shelf models pre-trained on large datasets. The extraction of scene and skeleton features follows the protocol outlined in Section 3.1.2.2. For Spatial features, we sample *n* equally-spaced frames from a given video and employ the InceptionResnet-v2 (SZEGEDY *et al.*, 2017), pre-trained on the Imagenet dataset, to extract a set of features $V_n = v_0, v_1, v_2, ..., v_n$, where n is the number of sampled frames. Then, an average pooling operation was performed into V_n, resulting in a final feature vector of dimension 1536. Following previous studies (WANG *et al.*, 2018c; DENG *et al.*, 2022), it was selected $n = 40$ frames. For temporal features, the I3D network (CARREIRA; ZISSERMAN, 2017), pre-trained on the Kinetics dataset (KAY *et al.*, 2017), was used to extract a 1024-dimensional feature vector of each video. Finally, a linear

projection is applied to the concatenated feature vector for dimensionality reduction purposes, resulting in a final vector of size 512.

The decoder is similar to that proposed by DEL CHIARO *et al.* (2020), which was inspired by the Hard Attention to the Task (HAT) approach (SERRA *et al.*, 2018) and extends the LSTM network by incorporating attention masks to overcome the catastrophic forgetting problem. The proposed approach differs from DEL CHIARO *et al.* (2020) in four main aspects: i) the proposed method is designed for the video captioning task and, therefore, considers video features as the input; ii) the proposed architecture is dynamically expanded to learn new knowledge from the new task; iii) We did not use the binary mask in the classifier; iv) during the inference time, the proposed method does not consider the task-ID and must, therefore, be capable of describing a given video with all available vocabulary. Figure 14 provides an overview of the proposed approach.

The proposed approach extends the LSTM network by incorporating two attention masks to control the use of network neurons for a given task. The computation of these attention masks is performed as follows:

$$\mathbf{m}_x = \sigma(s\mathbf{A}_x t^T) \tag{37}$$

$$\mathbf{m}_h = \sigma(s\mathbf{A}_h t^T) \tag{38}$$

where σ refers to the sigmoid activation function, t is a one-hot task vector, A_x and A_h are embedding matrices, and s is a positive scaling factor. The s parameter is linearly annealing during training, as suggested by Serra *et al.* (2018), and is defined as $s = \frac{1}{s_{max}} + (s_{max} - \frac{1}{s_{max}})\frac{b-1}{B-1}$, where b is the batch index and B is the total number of batch for the epoch. s_{max} was empirically defined as 400.

The final computation process in the decoder step, considering the attention masks, is defined by the equations presented below:

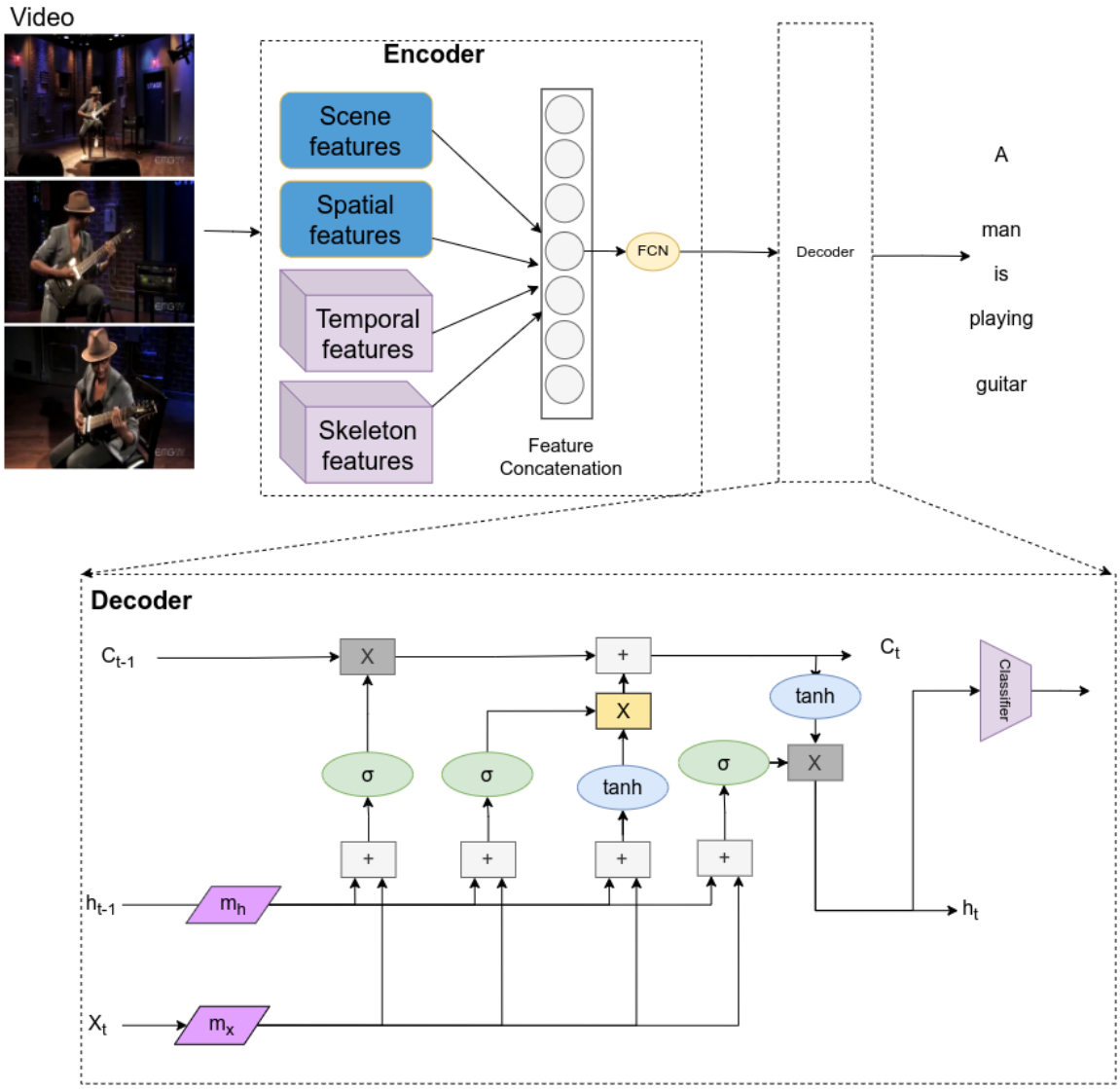
$$\mathbf{h}_t = tanh(\mathbf{C}_t) * \mathbf{o}_t, \tag{39}$$

$$\mathbf{C}_t = \sigma(\mathbf{f}_t * \mathbf{C}_{t-1} + \mathbf{i}_t * \tilde{\boldsymbol{C}}_t), \tag{40}$$

$$\tilde{\boldsymbol{C}}_t = tanh(\mathbf{m}_x \mathbf{x}_t \mathbf{U}^g + \mathbf{m}_h \mathbf{h}_{t-1} \mathbf{W}^g), \tag{41}$$

where σ and $tanh$ are the sigmoid and hyperbolic tangent activation function, respectively; $\mathbf{m}_x$ and $\mathbf{m}_h$ are attention masks for input and hidden state layers, respectively; $\mathbf{U}^g$ and $\mathbf{W}^g$ are weight

Figure 14 – Overview of the video captioning approach with an attention-based mechanism for preventing the catastrophic forgetting problem.



Source: Developed by the author.

matrices; $\mathbf{x}_t$ is the input at time t; $\mathbf{h}_{t-1}$ is the previous state; and $\mathbf{f}_t$, $\mathbf{i}_t$, and $\mathbf{o}_t$ are the forget, input and output gates, respectively. The calculations of unit gates are:

$$\mathbf{f}_t = \sigma(\mathbf{m}_x \mathbf{x}_t \mathbf{U}^f + \mathbf{m}_h \mathbf{h}_{t-1} \mathbf{W}^f + \mathbf{b}_f), \tag{42}$$

$$\mathbf{i}_t = \sigma(\mathbf{m}_x \mathbf{x}_t \mathbf{U}^i + \mathbf{m}_h \mathbf{h}_{t-1} \mathbf{W}^i + \mathbf{b}_i), \tag{43}$$

$$\mathbf{o}_t = \sigma(\mathbf{m}_x \mathbf{x}_t \mathbf{U}^o + \mathbf{m}_h \mathbf{h}_{t-1} \mathbf{W}^o + \mathbf{b}_o), \tag{44}$$

in which $\mathbf{U}^f$, $\mathbf{U}^i$, $\mathbf{U}^o$, $\mathbf{W}^f$, $\mathbf{W}^i$, and $\mathbf{W}^o$ are weight matrices, $\mathbf{b}_f$, $\mathbf{b}_i$ and $\mathbf{b}_o$ are bias vectors

3.2.2.1 Model Architecture Expansion

When training a new task t, the architecture must be expanded to incorporate new knowledge, including task-specific words. To accomplish this objective, a new Embedding layer E_{new} is created to encode both old and new words, considering the vocabulary introduced by the new task t. If the internal memory capacity of the decoder model is considered insufficient to accommodate new knowledge, a dynamic expansion takes place by assessing the availability of neurons for future tasks. When the number of available neurons falls below a certain threshold, denoted as k, the internal memory is increased by d. Here, k and d were empirically defined. When the internal memory is expanded, the masks m_h are also expanded. Finally, a new classifier C_{new} is created, considering the vocabulary introduced in the task. All expanded layer weights of model M_t are initialized with all shared model weights trained in task M_{t-1}. Algorithm 1 summarizes updating the model to train a new task.

Algorithm 1 – Model architecture expansion for training new tasks.

inserir $curVocab \leftarrow$ current task vocabulary list
inserir $oldVocab \leftarrow$ previous task vocabulary list
inserir $LSTM \leftarrow$ current decoder model
1: Set $newVocab$ as intersection of $curVocab$ and $oldVocab$
2: create new embedding layer E_{new} whit $newVocab$ size
3: initialize weights of E_{new} with shared weights of E
4: set k as a threshold value
5:
6: set $capacity$ to $1 - [m_h.sum()/m_h.len()]$
7: **se** $capacity < k$ **então**
8: create $LSTM_{new}$ with size increased by d
9: initialize weights of $LSTM_{new}$ with shared weights of $LSTM$
10: create new mask $m_{h_{new}}$ considering $LSTM_{new}.hidden\_size$
11: initialize weights of $m_{h_{new}}$ with shared weights of m_h
12: **finaliza se**
13:
14: create new classifier C_{new} with size of $newVocab$
15: initialize weights of C_{new} with shared weights of C

Source: Developed by the author.

3.2.2.2 Training Process

The attention masks used in the proposed method aim to preserve the knowledge accumulated by the network, conditioning the gradients according to the cumulative attention from all the previous tasks. Such masks are only considered in the backward computation process when task $t > 1$ is learned, as depicted in Figure 13. Therefore, before training a task $t > 1$, the

cumulative attention vector of the masks is recursively computed as follows:

$$\mathbf{m}_x^{<t} = max(\mathbf{m}_x^{t-1}, \mathbf{m}_x^{<t-1}) \tag{45}$$

$$\mathbf{m}_h^{<t} = max(\mathbf{m}_h^{t-1}, \mathbf{m}_h^{<t-1}) \tag{46}$$

During the network training process of a task $t > 1$, the gradient g of layer l is computed, taking into account the attention masks, as follows:

$$\mathbf{g}_{l,ij}^{'} = [1 - min(\mathbf{m}_{l,i}^t, \mathbf{m}_{l,j}^t)]\mathbf{g}_{l,ij} \tag{47}$$

where i refers to the i-th element of vector m_l at task t and j is the j-th element of vector m_l at task t; such a equation is applied in both m_x and m_h masks. Such a computation prevents large changes to the important weights for previous tasks.

Moreover, the gradient compensation procedure described in Serra *et al.* (2018) is applied to help train the task-embedding matrices A_x and A_h, as defined below.

$$\mathbf{q}_{l,ij}^{'} = \frac{s_{max}[cosh(sA_{x,i}t^T) + 1]}{s[cosh(A_{x,i}t^T) + 1]} q_{l,i} \tag{48}$$

for numerical stability, we clamp $|sA_{x,i}t^T| \leq 50$ and $|sA_{h,i}t^T| \leq 50$.

The model is trained using the cross-entropy loss, denoted as $\mathcal{L}_{CE}$, combined with a regularization term $\mathcal{L}_a$ to encourage low network usage and preserve the availability of some neurons for future tasks. The final loss is computed as follows:

$$\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_a = -\sum log(p_t^*|p_1^*,...,p_{t-1}^*, V) + \lambda \frac{\sum_i m_{x,i}^t (1 - m_{x,i}^{<t})}{\sum_i (1 - m_{x,i}^{<t})} + \lambda \frac{\sum_i m_{h,i}^t (1 - m_{h,i}^{<t})}{\sum_i (1 - m_{h,i}^{<t})} \tag{49}$$

where p^* refers to the probability of the ground truth word at the time step t, V is the video feature representation, and λ is a regularization constant that controls the capacity spent on each task. A larger λ promotes a reduced allocation of neurons to the current task, thereby increasing the number of remaining neurons available for future tasks. As suggested by DEL CHIARO *et al.* (2020), λ was set to 5000.

3.2.3 Evaluation Protocol

Three metrics are frequently employed when evaluating class-incremental learning approaches on classification tasks: average accuracy, forgetting, and intransigence (MASANA *et*

al., 2023). In this section, we use these metrics to report the conducted experiments. However, instead of assessing the model's performance using the accuracy measure, the METEOR metric was used. Such a decision was motivated by its widespread use in the literature for video captioning tasks since it considers morphological variations and synonyms. The METEOR metric was computed using the Common Objects in Context (COCO)-caption API (CHEN *et al.*, 2015)

Average accuracy (LOPEZ-PAZ; RANZATO, 2017) measures the overall performance of the model on all t tasks. After learning each task *t*, all classes seen so far are evaluated using the latest model. Thus, the $a_{t,k} \in [0,1]$ denotes the performance of task k after learning task t ($k \leq t$). The overall learning process can be measured by Equation 50 to compare the performance of different approaches with a single value. A higher A_t implies a better approach. Although the term "average accuracy" is used in this thesis, the performance evaluation employed the METEOR metric instead of conventional accuracy measurement in the classification task. The METEOR metric is widely employed in the literature for evaluating video captioning approaches.

$$A_t = \frac{1}{t}\sum_{i=1}^{t} a_{t,i} \tag{50}$$

Forgetting measure (CHAUDHRY *et al.*, 2018) estimates the model's forgetting rate concerning previous tasks after learning a new task. It is calculated by Equation 51, which considers the difference between the maximum performance obtained in the previous tasks and the performance in the current task.

$$f_j^k = \max_{l \in 1,\ldots,k-1} a_{l,j} - a_{k,j} \forall j < k \tag{51}$$

Note that $f_k^t \in [-1,1]$ quantifies forgetting and can be negative when the performance of task k after training task t is higher than obtained while training previous tasks. Equation 52 computes the average forgetting AF of all old tasks. A lower AF_k implies less forgetting on previous tasks.

$$AF_k = \frac{1}{k-1}\sum_{j=1}^{k-1} f_j^k \tag{52}$$

Intransigence measure (CHAUDHRY *et al.*, 2018) assesses the inability of a model to learn new tasks by comparing its performance against a reference model. The equation for the intransigence measure is defined as follows:

$$I_k = a_k^* - a_{k,k} \tag{53}$$

where a_k^* denotes the accuracy of the hold-out set of the k-th task on the reference model trained with dataset $\bigcup_{l=1}^{k} = D_l$, and $a_{k,k}$ denotes the performance of the model on the k-task trained incrementally; The I_k may range from -1 to 1, where a higher I_k indicates that the model do not perform well on new tasks. Notice that $I_k < 0$ implies a positive influence of previous tasks on the current task k. The average intransigence can be computed as follows:

$$AI_t = \frac{1}{t} \sum_{k=1}^{t} I_k \tag{54}$$

where I_k can be computed using the Equation 53.

This study did not consider other metrics proposed for incremental learning tasks, including Forward/Backward Transfer of knowledge (MAI *et al.*, 2022). Using such metrics in the class-incremental learning problem is not helpful as the increment of new classes in new tasks drops the performance and does not contribute to such metrics (MASANA *et al.*, 2023).

4 EXPERIMENTS, RESULTS AND DISCUSSION

This Chapter presents the experiments conducted to evaluate the proposed methods described in Section 3. The Chapter is divided into two Sections: the first presents the results of the method proposed in Section 3.1, which introduces an approach for describing known video events and recognizing unknown events. The second Section focuses on the method presented in Section 3.2 for video captioning within a class-incremental learning setting.

4.1 VIDEO DESCRIPTION METHOD IN AN OPEN-SET SCENARIO

In this Section, experiments were performed to evaluate the OSVidCap, as presented in Section 3.1. The main objective of such experiments is to evaluate the performance in describing concurrent known events in a given video and detecting unknown events. Also, the performed experiments aim to evaluate the influence of different features, such as the Human body skeleton and Place-type features, to understand fine-grained actions frequently performed in specific environments.

4.1.1 Datasets

There are a few datasets publicly available for video captioning task (AAFAQ *et al.*, 2019). The most used datasets in the literature are MSVD (CHEN; DOLAN, 2011) and MSR-VTT (XU *et al.*, 2016), containing a wide variety of open domain short videos. Each video has only a single main activity and multiple sentences with different details describing the video.

Despite the availability of annotated datasets for the video captioning task, none of them contains specific information about the action performed in each video, such as an action categorization. This information is essential in detecting and recognizing known and unknown events in an open-set scenario. Also, they do not contain concurrent events happening in the same video.

To overcome the above-mentioned limitations, we improved the LIRIS human activities dataset (WOLF *et al.*, 2014) with captions and temporal annotations of new actions. Furthermore, we evaluate the generalization of our method on the large-scale ActivityNet Captions dataset.

Both datasets are detailed as follows and are made available for further studies[1].

4.1.1.1 LIRIS human activities dataset

The LIRIS human activities dataset was proposed by Wolf *et al.* (2014) for recognizing complex and realistic actions in videos and was made available for the Human Activities Recognition and Localization (HARL) competition at the 2012 International Conference on Pattern Recognition (ICPR). The full dataset contains 828 actions (including discussing, telephone calls, giving an item, etc.) performed by 21 different people in 10 different classes. Each action performed in a video contains spatial annotations in a bounding box and temporal information (the beginning and end of the action). It was organized into two independent subsets: the D1 subset, with depth and grayscale images, and the D2 subset, with color images. The dataset also has unannotated actions, such as walking, running, whiteboard writing, book leafing, etc. The D2 subset contains 365 temporal annotated actions from 167 videos, and each action consists of one or more people performing one or more different activities.

We improved the dataset by providing natural language descriptions for the D2 subset of the LIRIS human activities dataset, allowing their use in the evaluation of video captioning. As stated by Vedantam *et al.* (2015), the number of reference sentences directly affects the accuracy of automated metrics. Also, those authors affirm that using five different sentences gives a substantial performance boost compared to only one sentence. Therefore, we provided five different descriptions for each video segment, as presented in Figure 15.

Table 3 presents the proposed dataset concerning the number of video segments and vocabulary size.

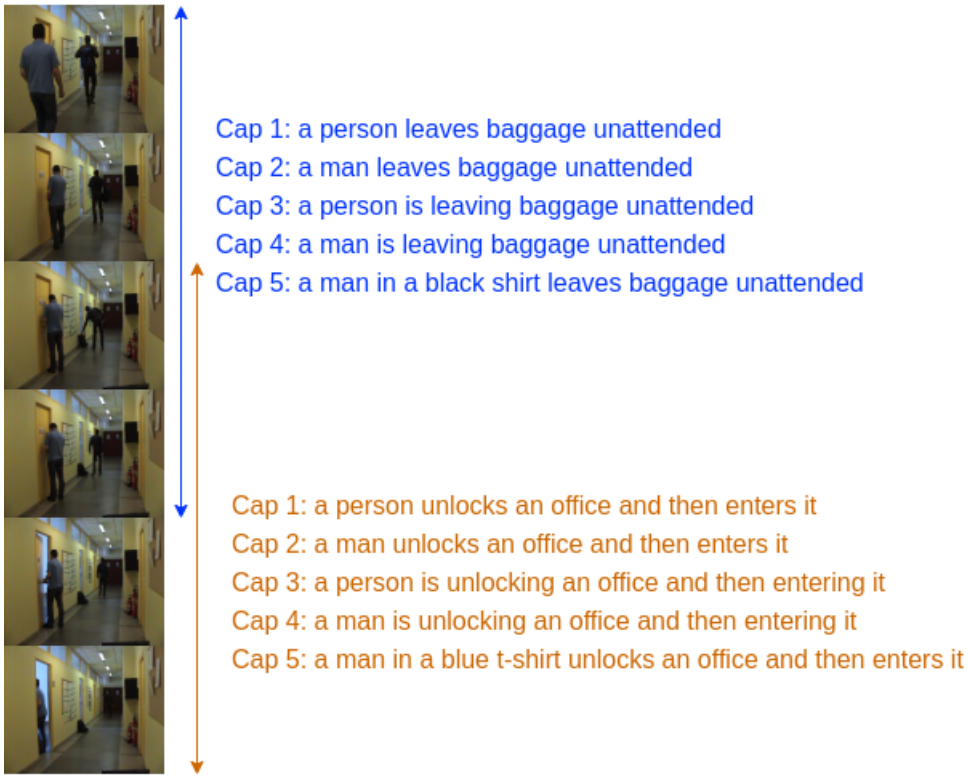
Table 3 – Overview of LIRIS dataset classes. Vocab. denotes the number of vocabulary; N.Videos denotes the number of videos.

N	Category	Vocab.	N. Videos
1	talks to second person	29	41
2	gives something to someone	24	20
3	puts/pick something	37	66
4	leave/enter place	27	91
5	try enter unsuccessfully	24	25
6	unlock door and enter it	28	24
7	leave something	23	23
8	shaking hands	25	39
9	typing / using computer	24	43
10	talks on the phone	24	23

Source: Developed by the author.

[1] http://labic.utfpr.edu.br/datasets/UTFPR-OSVidCAP.html

Figure 15 – Example of a video clip and the ground-truth sentences created for each human activity in the LIRIS human activities dataset. Blue and brown captions correspond to two different concurrent activities performed by different actors.



Source: Developed by the author.

Besides, we extracted 116 video segments in 15 different unannotated actions from the original videos to be used as unknown classes. Each new video segment was also annotated with spatial, temporal, and description information.

4.1.1.2 ActivityNet Captions dataset

The ActivityNet Captions dataset (KRISHNA *et al.*, 2017) is a large dataset proposed for dense-captioning events, which involves both detecting and describing events in a video.

It contains 20,000 videos split into around 50%, 25%, 25% for training, validation, and testing set, respectively. All videos were taken from the ActivityNet Dataset (HEILBRON *et al.*, 2015), a benchmark for video classification and detection, which covers 200 classes of activities. The dataset also has an overlap of 10% of the temporal descriptions, thus indicating the presence of concurrent events. Each video is annotated with a series of temporally localized descriptions.

Although the ActivityNet Captions dataset is available for download as a collection of YouTube video links, many of these videos are no longer available for download, as reported in

previous works (IASHIN; RAHTU, 2020), and only the pre-computed C3D features provided by the authors are not helpful in our experiments. Thus, we used 12,714 videos that were still available for download. Videos shorter than 3 seconds were disregarded due to the small number of extracted frames. As our approach focused on describing entire videos and not detecting a series of events, we used the ground-truth event proposals to extract 34,934 video clips for each temporarily localized description provided in the annotations.

While ActivityNet Captions was originally designed for dense video captioning, we adapt it to the open-set scenario by including action annotations to evaluate the generality of the proposed method in a large-scale dataset. Due to the considerable effort required to annotate each video clip manually, these annotations were collected from the ActivityNet dataset based on the video name, which is the same in both datasets. Each resulting action class contains, on average, 114 videos for training and 55 videos for testing. Table 4 summarizes the ActivityNet Captions used in this study. The action annotations were used to split videos into known and unknown classes to detect known and unknown actions.

Table 4 – Overview of ActivityNet Captions dataset classes. Vocab. denotes the number of vocabulary; N.Videos denotes the number of videos.

#	Category	Vocab.	N. Videos
1	Applying sunscreen	422	125
2	Archery	480	172
3	Arm wrestling	576	209
4	Assembling bicycle	427	158
5	Baking cookies	542	207
6	Ballet	480	150
7	Bathing dog	332	118
8	Baton twirling	599	303
9	Beach soccer	462	172
10	Beer pong	572	205
11	Belly dance	378	157
12	Blow-drying hair	365	133
13	Blowing leaves	417	173
14	BMX	594	244
15	Braiding hair	424	166
16	Breakdancing	532	184
17	Brushing hair	334	124
18	Brushing teeth	341	114
19	Building sandcastles	458	168
20	Bullfighting	387	106
21	Bungee jumping	446	124
22	Calf roping	408	212
23	Camel ride	420	174
24	Canoeing	592	244
25	Capoeira	573	240
26	Carving jack-o-lanterns	402	139
27	Changing car wheel	518	173

N	Category	Vocab.	N. Videos
28	Cheerleading	597	213
29	Chopping wood	411	175
30	Clean and jerk	444	201
31	Cleaning shoes	397	129
32	Cleaning sink	378	128
33	Cleaning windows	427	121
34	Clipping cat claws	345	149
35	Cricket	391	123
36	Croquet	457	161
37	Cumbia	446	175
38	Curling	503	170
39	Cutting the grass	515	207
40	Decorating the Christmas tree	385	155
41	Disc dog	500	284
42	Discus throw	499	211
43	Dodgeball	443	169
44	Doing a powerbomb	362	139
45	Doing crunches	361	111
46	Doing fencing	527	183
47	Doing karate	471	152
48	Doing kickboxing	463	170
49	Doing motocross	471	162
50	Doing nails	577	182
51	Doing step aerobics	439	192
52	Drinking beer	314	87
53	Drinking coffee	430	108
54	Drum corps	473	193
55	Elliptical trainer	412	158
56	Fixing bicycle	509	182
57	Fixing the roof	469	155
58	Fun sliding down	353	164
59	Futsal	591	292
60	Gargling mouthwash	312	88
61	Getting a haircut	490	212
62	Getting a piercing	392	145
63	Getting a tattoo	452	152
64	Grooming dog	507	190
65	Grooming horse	454	172
66	Hammer throw	465	181
67	Hand car wash	475	208
68	Hand washing clothes	417	137
69	Hanging wallpaper	586	190
70	Having an ice cream	428	145
71	High jump	454	173
72	Hitting a pinata	527	212
73	Hopscotch	432	176
74	Horseback riding	551	188
75	Hula hoop	465	128
76	Hurling	578	249
77	Ice fishing	400	149
78	Installing carpet	461	142
79	Ironing clothes	319	106
80	Javelin throw	389	158
81	Kayaking	532	214
82	Kite flying	354	127
83	Kneeling	483	209
84	Knitting	343	136

N	Category	Vocab.	N. Videos
85	Laying tile	456	138
86	Layup drill in basketball	382	141
87	Long jump	473	168
88	Longboarding	700	307
89	Making a cake	780	278
90	Making a lemonade	549	254
91	Making a sandwich	523	178
92	Making an omelette	359	126
93	Mixing drinks	468	224
94	Mooping floor	426	175
95	Mowing the lawn	494	190
96	Paintball	542	178
97	Painting	460	171
98	Painting fence	419	164
99	Painting furniture	351	105
100	Peeling potatoes	300	95
101	Ping-pong	467	145
102	Plastering	461	170
103	Plataform diving	442	161
104	Playing accordion	391	149
105	Playing badminton	374	153
106	Playing bagpipes	446	171
107	Playing beach volleyball	370	128
108	Playing blackjack	401	153
109	Playing congas	550	301
110	Playing drums	505	250
111	Playing field hockey	483	186
112	Playing flauta	381	136
113	Playing guitarra	445	205
114	Playing harmonica	459	168
115	Playing ice hockey	488	179
116	Playing kickball	352	140
117	Playing lacrosse	409	161
118	Playing piano	346	125
119	Playing polo	478	182
120	Playing pool	461	181
121	Playing racquetball	341	152
122	Playing rubik cube	364	161
123	Playing saxophone	440	174
124	Playing squash	430	172
125	Playing ten pins	508	208
126	Playing violin	496	208
127	Playing water polo	530	232
128	Pole vault	534	191
129	Polishing forniture	401	107
130	Polishing shoes	380	132
131	Powerbocking	609	230
132	Preparing pasta	569	251
133	Preparing salad	527	202
134	Putting in contact lenses	380	148
135	Putting on makeup	466	190
136	Putting on shoes	327	108
137	Rafting	628	309
138	Raking leaves	317	109
139	Removing curlers	361	100
140	Removing ice from car	445	165
141	Riding bumper cars	398	230

N	Category	Vocab.	N. Videos
142	River tubing	548	233
143	Rock climbing	496	146
144	Rock-paper-scissors	342	98
145	Rollerblading	624	213
146	Roof shingle removal	447	138
147	Rope skipping	615	264
148	Running a marathon	536	190
149	Sailing	515	188
150	Scuba diving	596	244
151	Sharpening knives	482	189
152	Shaving	407	156
153	Shaving legs	378	141
154	Shot put	472	173
155	Shoveling snow	462	170
156	Shuffleboard	459	175
157	Skateboarding	504	153
158	Skiing	628	257
159	Slacklining	544	257
160	Smoking a cigarette	292	109
161	Smoking hookah	344	114
162	Snatch	502	164
163	Snow tubing	510	245
164	Snowboarding	583	245
165	Spinning	432	178
166	Spread mulch	323	90
167	Springboard diving	382	165
168	Starting a campfire	461	162
169	Sumo	494	154
170	Surfing	506	274
171	Swimming	455	159
172	Swinging at the playground	413	184
173	Table soccer	446	164
174	Tai chi	490	163
175	Tango	518	217
176	Tennis serve with ball bouncing	365	140
177	Throwing darts	465	205
178	Trimming branches or hedges	442	166
179	Triple jump	504	206
180	Tug of war	470	201
181	Tumbling	460	158
182	Using parallel bars	538	276
183	Using the balance beam	533	297
184	Using the monkey bar	388	167
185	Using the pommel horse	417	197
186	Using the rowing machine	381	156
187	Using uneven bars	362	142
188	Vacuuming floor	397	146
189	Volleyball	402	132
190	Wakeboarding	627	247
191	Walking the dog	520	183
192	Washing dishes	493	195
193	Washing face	414	150
194	Washing hands	425	148
195	Waterskiing	582	244
196	Waxing skis	493	157
197	Welding	445	159
198	Windsurfing	416	180

N	Category	Vocab.	N. Videos
199	Wrapping presents	513	199
200	Zumba	412	147

Source: Developed by the author.

4.1.2 Implementation Details

The proposed OSVidCap framework uses an encoder-decoder architecture. Therefore, both the encoder and caption generation (decoder) modules were trained end-to-end. Before training, all captions were tokenized and converted to lowercase. Sparse words occurring less than three times in the training set were replaced with the unknown token. The *fasttext* (BOJANOWSKI *et al.*, 2017) word embedding pre-trained on the Common Crawl Corpus was used to embed features into a 300-dimensional feature vector. It provides much more powerful and effective low-dimensional word representations for video captioning than other techniques, such as sparse one-hot encoding vectors (AAFAQ *et al.*, 2021).

During the training step, a begin-of-sentence and end-of-sentence token were added to the sentence to deal with varying lengths. Also, an unknown tag was used to replace sparse words. We input the begin-of-sentence token into our Caption Generation Module to start the description generation process during the test step. Then, previously generated words are used as input to produce the following words until the max sentence length or the end-of-sentence token is achieved. In our experiments, the max sentence length was set as 19 and 25 for the LIRIS and ActivityNet Captions dataset, respectively. Zero padding is applied if the sentence is shorter than the maximum number of words. The Beam Search method was employed to select the best sentence and avoid local optima. In our experiments, the beam size k was set to 3.

We empirically set the hidden state LSTM with 512 units and applied dropout with a rate of 0.5 on the input and output of the LSTM. The Adam algorithm, with a learning rate of 5×10^{-5} was used for optimization. The cross-entropy loss was used to train our model. All experiments were implemented using Tensorflow[2] and Keras[3] library.

To demonstrate the effectiveness of the proposed method, we have conducted experiments on two datasets (LIRIS human activities and ActivityNet Captions dataset) to analyze the influence of the open set module and compare the video caption performance with related works.

[2] https://www.tensorflow.org/
[3] https://keras.io/

Due to the small number of videos and known actions in the LIRIS dataset, we performed a 5-fold cross-validation procedure to assess the OSVidCap performance. The same training and testing set of each cross-validation fold was used to train the open set module. In addition, to evaluate the effectiveness of the proposed approach in detecting unknown events, the testing set included 116 videos with unknown actions, as described in Section 4.1.1.1.

The performance of OSVidCap for known events on the ActivityNet Captions Dataset was performed using the standard data split[4]. Since this dataset was made available as a challenge, the test set was not provided with the ground truth. Thus, we follow the previous works (IASHIN; RAHTU, 2020; ZHOU *et al.*, 2018b) and report the results on the validation set. The effectiveness of the proposed approach in detecting unknown events was also performed using a 5-fold cross-validation procedure. Each fold contains known videos of 40 actions for the training and testing set, as explained in Section 4.1.1.2. We also included v_r random videos from other classes as unknown actions in the testing set. The v_r was defined as the same number of videos presented in the training set to avoid imbalanced data.

4.1.3 Quantitative Results

In this section, the performance evaluation of the proposed method is presented and compared with two existing approaches.

Semantic Grouping Network (SGN) (RYU *et al.*, 2021) exploits the use of semantic groups based on meanings such as people, objects, or actions rather than frame by frame for understanding a video. It is comprised of four main components: (i) a Visual Encoder component that aims to extract visual features from video frames; (ii) a Phrase Encoder that produces phrase representations from words by using the self-attention mechanism; (iii) a Semantic Grouping which employs a semantic aligner to align the video frames with phrases; and (iv) a Decoder based on LSTM with temporal attention.

Non-Autoregressive Coarse to-Fine (NACF) model (YANG *et al.*, 2021a) proposes a coarse-to-fine captioning procedure using a bi-directional self-attention-based network as caption generator. For improving caption quality, the decoder method is decomposed into two stages. First, a coarse-grained "template" is generated. Then, dedicated decoding algorithms generate fine-grained descriptions by filling in the generated "template" with suitable words and modifying inappropriate phrasing via iterative refinement.

[4] https://cs.stanford.edu/people/ranjaykrishna/densevid/

For a fair comparison, all the methods utilize the ResNet-101 and ResNext-101 features as input, and the reported results were obtained using Microsoft COCO caption evaluation tool (CHEN *et al.*, 2015). Furthermore, all approaches were set with the same maximum sentence length and minimum word frequency during training.

Table 5 presents a comparison performance of the OSVidCap with existing approaches on LIRIS human activities dataset. It can be noticed that our model OSVidCap$_{(S+T)}$ achieved better performance in terms of ROUGE-L and CIDEr and competitive performance in terms of BLEU and METEOR. Also, our model OSVidCap$_{(S+T+SK+P)}$ surpasses the compared approaches by 4.9% of BLEU-4, 5.1% of METEOR, 4.3% of ROUGE-L, and 9.3% of CIDEr. This suggests that our approach can better describe concurrent events in videos. In addition to spatial (S) and temporal (T) features, the model considered Human body skeleton (SK) extracted from human movements and Place-Type (P) features extracted from places. This points out that specialized features can be essential to better describe similar actions or actions according to the context (place). Such feature enrichment provides essential information to distinguish some actions, such as shaking hands and giving a small item to a second person. Also, the place type gives meaningful semantic information, as some actions tend to happen in specific places.

Table 5 – Comparison Performance of video captioning on the LIRIS human activities dataset. 5-fold cross-validation results are presented in terms of BLEU-4 (B), METEOR (M), ROUGE-L(R), and CIDEr (C). S denotes Spatial features. T denotes temporal features. SK denotes Skeleton features. P denotes Place features.

Model	B	M	R	C
NACF (YANG *et al.*, 2021a)	66.27	46.94	80.52	323.66
SGN (RYU *et al.*, 2021)	62.08	44.38	76.95	298.06
OSVidCap $_{(S+T)}$	65.28	46.49	80.69	330.65
OSVidCap $_{(S+T+SK)}$	69.50	49.31	**84.05**	351.19
OSVidCap $_{(S+T+SK+P)}$	**69.54**	**49.34**	83.78	**354.04**

Source: Developed by the author.

Table 6 presents the video captioning comparison on the ActivityNet Captions dataset. It can be noticed that the proposed approach also achieved better or competitive results across all metrics, showing robust generalization to other contexts and scenarios. It is also noteworthy that the values of the metrics presented in Table 6 are significantly lower than those presented in Table 5 due to the complexity of the datasets, as reported in Section 4.1.1.2. The performance reported on this dataset is similar to those reported in recent literature (IASHIN; RAHTU, 2020; DENG *et al.*, 2021). Note that, despite using the same dataset to report the results, they are not comparable with the presented approach, as the videos and features used for training, validation, and testing are different.

Table 6 – Video Captioning Performance on the ActivityNet Captions validation set. Results are presented in terms of BLEU-4 (B), METEOR (M), ROUGE-L(R), and CIDEr (C). S denotes Spatial features, T denotes temporal features, SK denotes Skeleton features, and P denotes Place features.

Model	B	M	R	C
NACF (YANG *et al.*, 2021a)	2.20	8.16	20.44	23.78
SGN (RYU *et al.*, 2021)	3.90	9.72	20.38	29.69
OSVidCap $_{(S+T)}$	4.19	9.71	20.98	28.54
OSVidCap $_{(S+T+SK)}$	4.30	9.96	21.26	**30.50**
OSVidCap $_{(S+T+SK+P)}$	**4.32**	**9.98**	**21.33**	29.84

Source: Developed by the author.

In both datasets, the use of Place-type features did not show significant improvements. This may indicate that previously used features can also describe this visual information or are irrelevant to the video description task.

In Table 7, one can observe the evaluation performance of the open-set module in detecting known and unknown actions on the LIRIS human activities Dataset. Results are presented in a 5-fold cross-validation procedure. The proposed method achieved satisfactory results in detecting known and unknown classes with an average F1-Score of 86.2%.

Table 7 – Open-Set Module on LIRIS Captions dataset.

		F1-Score		Precision		Recall	
		Unknown	Known	Unknown	Known	Unknown	Known
1	89.0%	92.0%	85.0%	86.00%	100.0%	100.0%	74.0%
2	86.0%	90.0%	81.0%	84.0%	96.0%	98.0%	70.0%
3	83.0%	90.0%	77.0%	81.0%	100.0%	100.0%	63.0%
4	88.0%	92.0%	84.0%	85.0%	100.0%	100.0%	76.0%
5	85.0%	91.0%	80.0%	83.0%	100.0%	100.0%	67.0%
AVG	86.2%	91.0%	81.4%	83.8%	99.2%	99.6%	70.0%

Source: Developed by the author.

Table 8 shows the evaluation performance of the open-set module in detecting known and unknown actions on the ActivityNet Captions dataset. Five experiments with different numbers of the known classes in a cross-validation procedure were performed. The proposed method achieved satisfactory results in detecting known and unknown classes with an average F1-Score of 79.80% when ten classes were considered as known actions.

Table 8 – Open-Set Module on ActivityNet Captions dataset.

Known classes	k-Fold	Average F1-Score	Average F1-Score		Average Precision		Average Recall	
			Unknown	Known	Unknown	Known	Unknown	Known
10	20	79.80%	79.20%	80.60%	84.10%	76.95%	75.15%	85.15%
20	10	77.10%	76.20%	78.10%	81.40%	74.00%	71.80%	82.90%
25	8	75.50%	75.38%	76.30%	78.63%	73.75%	69.00%	82.63%
40	5	73.60%	72.60%	74.60%	77.20%	71.00%	68.60%	79.00%
50	4	72.50%	71.25%	73.75%	75.50%	69.50%	67.50%	77.50%

Source: Developed by the author.

It can also be seen in Table 8 that the average precision of the unknown class is about 9% higher than the known class, and the average recall of the known class is 13% higher than the unknown class. This shows that the proposed approach achieves better results in detecting unknown classes than known classes. The automatic annotation process of video actions on the ActivityNet Captions dataset, as described in Section 4.1.1.2, also produced some annotation noises during the training and testing process. These noises can be considered a performed action with a different label or even a video without human actions.

Figure 16 depicts an example of a video presented in the dataset. It can be observed that the video has different events with different start and end times. The automatic annotation process set the action class "Removing ice from car" to all video clips. However, in this example, only two video clips are related to the annotated action. Therefore, the degradation in the average precision metric of the known class may have been caused by the presence of these annotation noises. When considering new actions as known classes, the average F1-Score decreased due to the cumulative annotation errors provided by the automatized annotation process, as reported below.

Table 9 reports the impact of the open-set component on the video descriptions generated by the proposed approach. The results reported in the LIRIS dataset used the same data in a cross-validation procedure as used to report the results in Table 7. For reporting the results on the ActivityNet Captions dataset, we also used the 5-fold cross-validation applied in Table 8.

Table 9 – Influence of the open set module in the OSVidCap approach. S denotes Spatial features. T denotes temporal features. SK denotes Skeleton features. P denotes Place features.

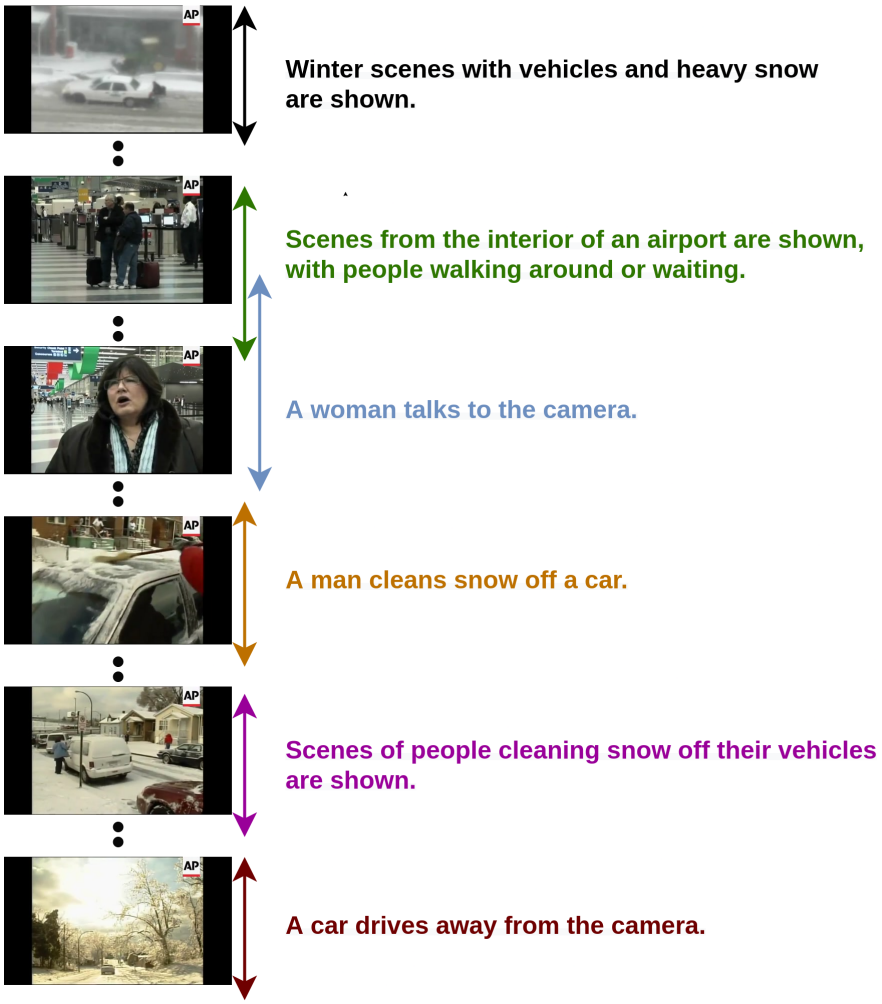
Dataset	Model	B	M	R	C	F1-score
LIRIS dataset	OSVidCap $_{(S+T)}$	77.8	56.1	87.9	381.1	86.2%
	OSVidCap $_{(S+T+SK)}$	**80.9**	**57.3**	**89.1**	385.8	
	OSVidCap $_{(S+T+SK+P)}$	80.2	57.2	**89.1**	**385.9**	
ActivityNet Captions dataset	OSVidCap $_{(S+T)}$	16.2	16.8	38.7	68.0	73.6%
	OSVidCap $_{(S+T+SK)}$	15.6	16.8	39.1	70.0	
	OSVidCap $_{(S+T+SK+P)}$	**16.5**	**16.9**	**39.2**	**72.9**	

Source: Developed by the author.

These results are significantly higher when compared with those reported in Tables 5 and 6 because, in this experiment, we considered videos in the test set with unknown activities. For these videos, the model is supposed to generate descriptions such as "a person is performing an unknown action".

The experiments with unknown actions in the testing set suggested that Place-type features did not lead to a significant improvement. However, these features are important to

Figure 16 – Example of events temporally localized in the video with independent start and end times, resulting in some events occurring concurrently in the ActivityNet Captions dataset.



Source: Developed by the author.

understand scenes in which the information about the place type is relevant, for example, to describe whether the person is entering or leaving an office or writing a whiteboard in a classroom. In the testing set used to report the experiments in Table 9, several videos from unknown classes were included to evaluate the proposed open set module. Therefore, the overall influence of the Place-type features has quantitatively decreased due to the small number of sentences that require such features. To the best of our knowledge, this is the first work to address the video captioning task in an open-set world by generating captions of known events present in the training set and dealing with unknown events not previously seen.

4.1.4 Qualitative Results

In Figure 17, we illustrate three examples of video descriptions generated by the baselines method SGN and NACF and the proposed OSVidCap. Figure 17a depicts a scene with two sequential actions. First, a man in a striped t-shirt talks to a woman in front of a whiteboard. Then, another man in a black t-shirt enters the room and gives an item to the man in a striped t-shirt. Figure 17b shows two concurrent events. While a man and a woman are handshaking, another man is leaving baggage unattended. Finally, in Figure 17c, three events take place in the video. At the same time, a man is performing an unknown action. Another man leaves an item in the letterbox cabinet and then enters the room.

For the examples of Figure 17, our approach described concurrent actions better than the baselines. In Figure 17a, the OSVidCap correctly described the ongoing action but wrongly represented the color of the t-shirt, suggesting that the model did not learn this information from the input features. Possibly, more specific features should fix this issue.

In Figure 17b, we can observe that the compared approaches could not detect the shake hands action, suggesting the importance of using human body features in describing human action videos. Also, they fail to detect and describe concurrent actions in videos.

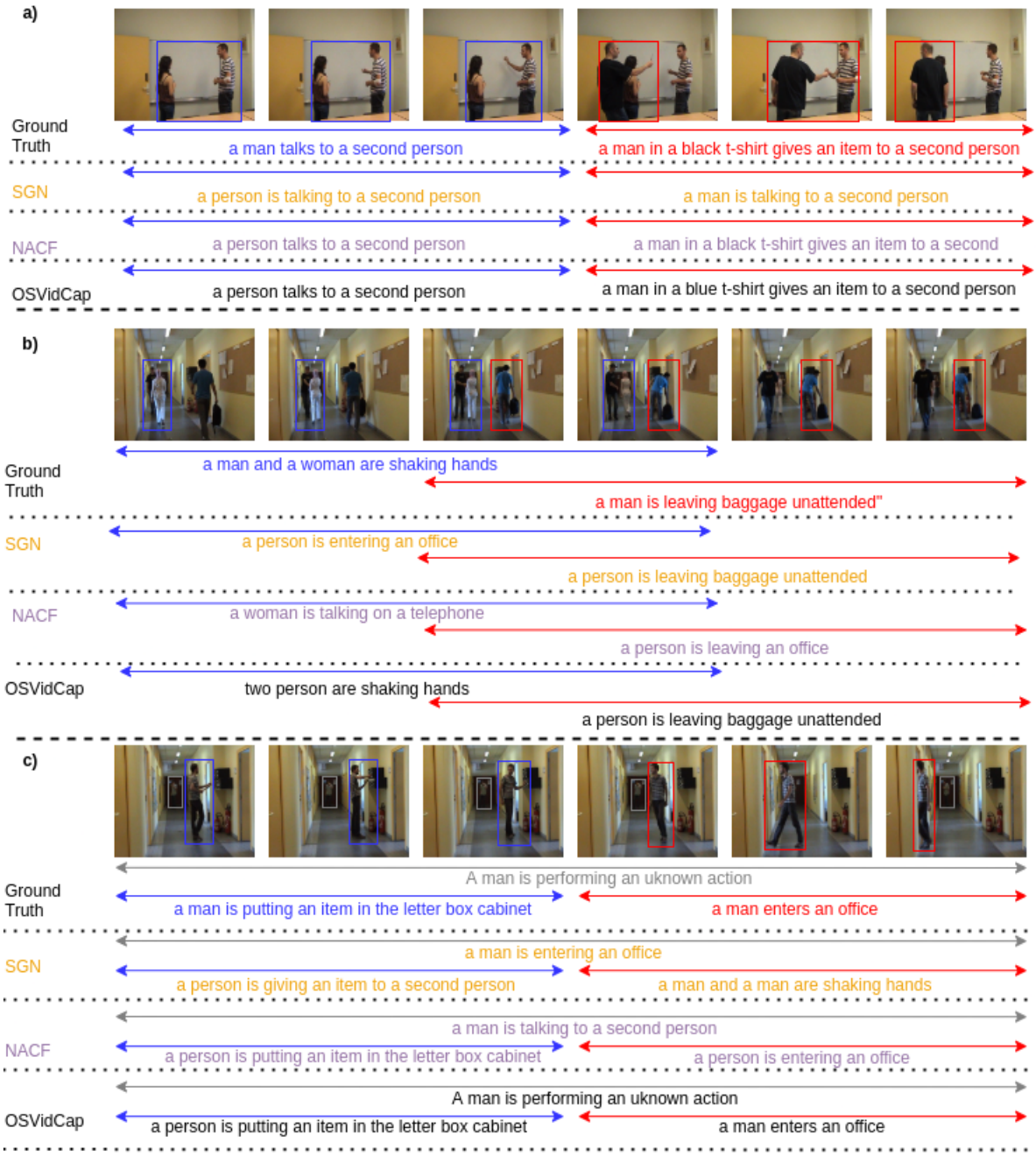
We can realize the importance of the open set module in the situation considered in Figure 17c). While the OSVidCap detected an unknown action performed by a man and correctly described it as such, the compared approaches generated a wrong description. It is worth highlighting that this action was previously labeled as unknown and did not appear in the training set.

4.1.5 Discussion

Most artificial intelligence methods for video captioning rely on the closed-set world assumption. Existing methods based on a closed-set world can adequately describe only the temporal events previously seen during the training step. Unless they are trained with all existing events and actions of interest, they will not be able to recognize unknown events found in videos in the wild. Furthermore, most current video description approaches focus only on single actions occurring at a time, while concurrent events may take place in the real world.

The experimental results presented in this section provide evidence of the effectiveness of the proposed framework in accurately describing concurrent events within a given video, as

Figure 17 – Qualitative example of generated descriptions on the LIRIS dataset.



Source: Developed by the author.

well as detecting unknown events. Furthermore, the findings highlight the importance of using different features as input, such as the Human body skeleton and Place-type features, which prove to be highly relevant for understanding fine-grained actions that are often performed in specific environments.

Despite the excellent results achieved by the OSVidCap, we observed that it could provide a more detailed description of people, including the type and color of the clothes. This enrichment of details can play an important role in surveillance applications. The TDL module

could detect individual humans or objects of interest and simple interactions between them by capturing the overlapping region among objects. However, the proposed module may fail to capture more complex human interactions.

The video captioning task in the open-set scenario is an open problem and not well-explored in the literature. The lack of appropriate datasets for the video captioning task in an open-set scenario may limit the development of new approaches. To address this limitation, new annotations of unknown actions have been contributed to the LIRIS human activities dataset. These annotations may serve as a valuable benchmark for the video captioning task in an open-set scenario, providing an alternative to evaluate and compare performance between state-of-the-art approaches. Moreover, a split for the ActivityNet Captions dataset was also proposed, considering the video description task in an open-world scenario.

4.2 VIDEO DESCRIPTION IN A CLASS-INCREMENTAL LEARNING SETTING

This section presents a series of experiments conducted to assess the efficacy of the proposed video description method in learning previously detected unknown video events using a class-incremental learning approach, as presented in Section 3.2. First, we provide a brief description of the datasets employed in these experiments, along with implementation details. Subsequently, we present the experiments organized into five subsections, each focusing on a research question that motivated its implementation, followed by an in-depth discussion of the findings.

4.2.1 Datasets

A class-incremental learning evaluation protocol involves splitting up a dataset with a set of tasks, where each task corresponds to a set of classes that are disjoint from the classes in other tasks, whether previous or future (MASANA *et al.*, 2023). Such a split is feasible in datasets for the classification task as the predicted classes are a single label.

Splitting an existing dataset for video captioning into disjoint tasks is not feasible due to a significant amount of shared vocabulary across different classes. This includes repeated usage of pronouns, articles, and nouns. To evaluate video captioning approaches within a class-incremental learning scenario, it is essential for each task to introduce novel and distinct words that are unseen in previous tasks. To the best of our knowledge, no datasets are available with

such characteristics for training video captioning approaches in a class-incremental learning scenario.

For this purpose, we use the proposed LIRIS human activities dataset, which is presented in Section 4.1.1.1, as a controlled dataset. The videos in this dataset are classified into ten distinct classes, representing different human actions. Each class contains specific words related to the corresponding task, allowing their use in a class-incremental learning scenario.

Although the LIRIS dataset can be used to assess the performance of the proposed method, a larger and open dataset is essential for generalization assessment. Using existing larger datasets for the video captioning task (such as MSR-VTT and ActivityNet Captions) is challenging due to the vocabulary overlap between classes and the high variability of actions and scenes. Therefore, we create a new dataset based on the MSR-VTT dataset, called here as MSR-VTT-subset, described in the following subsection.

4.2.1.1 MSR-VTT-subset dataset

The MSR-VTT Dataset is widely used to evaluate the performance of video captioning approaches, as described in Section 2.8.2. Although it covers a comprehensive list of 20 categories, it is not feasible to split the videos into tasks for training a video captioning approach in a class-incremental learning setting due to word overlap among categories. For example, a video categorized in the music category contains content from a music video clip with different associated reference sentences, such as “a girl band singing a song and playing musical instruments” and “a man on a motorcycle talking to a woman”. Despite the differences in the semantic meaning of the sentences, both can be considered correct and used to describe the video. The first sentence considers both audio and visual information, describing the entire video, while the second one focuses on the visual information of a small video segment.

As discussed previously, a dataset with disjoint vocabulary among tasks is required for testing video captioning approaches in a class-incremental learning scenario. Therefore, to address this issue, we created a subset of the MSR-VTT dataset, called MSR-VTT-subset, consisting of 20 action categories, each containing some disjoint words for its respective task. Such a dataset was created by filtering videos from the original dataset based on captioning words representing the scene. Table 10 presents a list of categories along with their corresponding vocabulary and the number of videos available. It is important to note that the dataset is imbalanced, which reflects the distribution of activities or events in the real world. This imbalance is

a common characteristic in many real-world applications, where certain activities occur more frequently or have more available data compared to others.

Table 10 – Overview of MSR-VTT-subset dataset classes. Vocab. denotes the number of vocabulary; N.Videos denotes the number of videos.

N	Category	Vocab.	N. Videos
1	playing baseball	862	58
2	cooking	1678	252
3	cutting/slicing	508	25
4	car driving	1447	143
5	dancing	1938	286
6	drinking	800	48
7	playing guitar	785	95
8	gun	819	50
9	kissing	648	39
10	make up	1121	99
11	minecraft	567	37
12	motorcycle	889	65
13	playing piano	396	35
14	playing basketball	432	29
15	playing videogame	918	91
16	rinding a horse	520	29
17	soccer	914	94
18	stroller	446	58
19	trampoline	274	27
20	wrestling	847	114

Source: Developed by the author.

4.2.2 Implementation Details

Before training the proposed approach, all captions were tokenized and converted to lowercase. Sparse words occurring less than three times in the training set were replaced with the unknown token. During the training step, a begin-of-sentence and end-of-sentence token was added to each sentence to handle varying lengths.

During the testing phase, the video content was fed into the Caption Generation Module to initiate the description generation process. Previously generated words were then used as input to generate subsequent words until the maximum sentence length or the end-of-sentence token was reached. In our experiments, the maximum sentence length was set as 19 and 20 for the LIRIS dataset and MSR-VTT-subset dataset, respectively. Zero padding was applied if a sentence was shorter than the maximum number of words. In the experiments conducted in this study, each task was considered a video category of the dataset.

The hidden state of the LSTM was empirically set to 1024. The Adam algorithm was employed for optimization, using a learning rate of 4×10^{-4}. All experiments were implemented

using the PyTorch[5] library and trained on a TITAN-XP GPU.

4.2.3 Experiment 1: Comparison between Approaches

The experiment presented in this section was motivated by the following question: "How does the proposed class-incremental learning technique compare to other techniques in terms of performance and forgetting, specifically considering its ability to retain previously learned knowledge while accommodating new classes?".

In such an experiment, it was set $k = 0$ in the Algorithm 1. Consequently, the model expansion process solely involves updating the input and output layers to incorporate new vocabulary. The impact of internal memory expansion was evaluated in Section 4.2.7.

The proposed method was compared with two recurrent network-based continual learning methods provided by DEL CHIARO *et al.* (2020), which facilitated the comparison with existing methods. Moreover, two baseline methods were implemented as lower and upper bounds, as suggested by De LANGE *et al.* (2022), and are described as follows.

- **Finetune**: It is a technique used in machine learning to adapt a pre-trained model to a new task or dataset. In a class-incremental learning scenario, this technique involves taking a pre-trained model from task t-1 and training it on a new task without any regularization technique. This approach is considered the lower bound of performance as it trains each task without implementing any strategy to prevent catastrophic forgetting, representing the minimum desired performance.

- **EWC**: Such an approach estimates the importance of each weight in a neural network using the Fisher information matrix, which quantifies how much each weight influences the loss function. By incorporating a penalty term based on the importance estimates into the loss function during training, EWC tries to selectively preserve important weights while allowing others to be more adaptable. More details about the method can be seen in (KIRKPATRICK *et al.*, 2017).

- **Recurrent Learning without Forgetting (Rec-LwF)**: It employs knowledge distillation on the LSTM decoder, involving both the teacher and student networks. At each decoding step, the teacher network (representing the decoder of the previous tasks) and the student

[5] https://pytorch.org/

network (representing the decoder of the new task) are provided with data from the new task and share a common word embedding. The output probabilities from the student network LSTM are compared with those predicted by the teacher network. A distillation loss ensures the student network does not deviate from the teacher. This process focuses on mitigating catastrophic forgetting, enabling the student network to preserve previously acquired knowledge while adapting to the demands of the new task, thereby facilitating the continuity of learning. More details about the distillation loss can be found in DEL CHIARO *et al.* (2020).

- **Joint**: This baseline combines the training data from all tasks and optimizes a single model to learn useful representations for all tasks. While it allows the network to learn shared representations that capture common patterns and features across different tasks, it may face challenges in finding a balance between optimizing for shared representations and maintaining task-specific performance. This baseline can provide a reference performance target (upper bound).

Table 11 shows the average accuracy of the proposed method on the LIRIS dataset. It is worth mentioning that, as described in Section 3.2.3, despite using the term 'average accuracy,' the METEOR metric was employed instead of accuracy as it is widely used to evaluate the performance of video description systems. The proposed method achieved superior scores in 60% of the tasks and, on average, outperformed the upper-bound model (Joint). Furthermore, the proposed model did not reach any scores lower than the Finetune, which serves as the lower bound. Compared to the Finetune approach, EWC consistently achieved equal or superior scores in the first six tasks, and the Rec-LWF demonstrated better performance during the intermediate tasks. Nevertheless, the Finetune approach achieved a similar average score due to its higher performance on the final task.

Table 11 – Average accuracy after learning the last task on the LIRIS dataset.

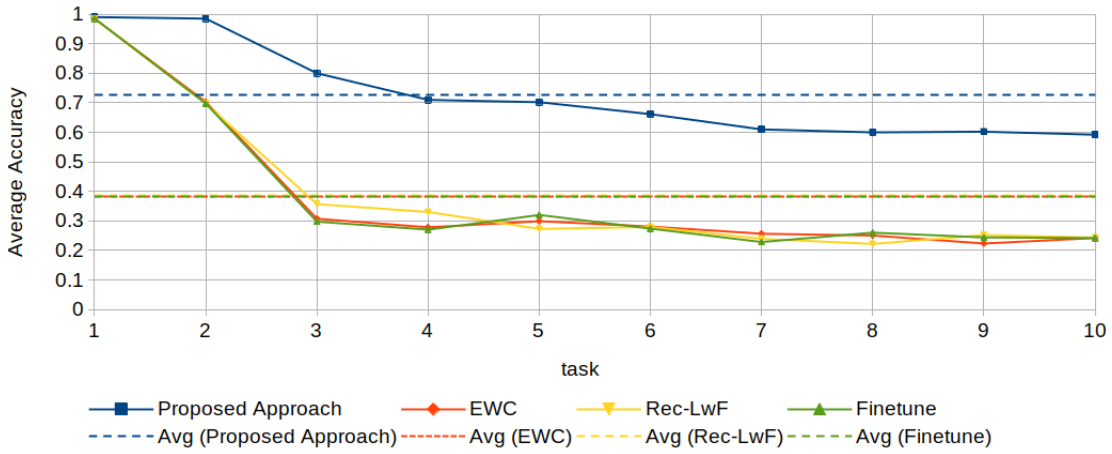
Model/task	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg**
Finetune	0.29	0.16	0.19	0.24	0.13	0.17	0.20	0.25	0.30	0.50	0.24
EWC	0.29	0.21	0.19	0.25	0.14	0.17	0.18	0.23	0.35	0.43	0.24
Rec-LwF	0.20	0.20	0.18	0.36	0.29	0.24	0.27	0.30	0.23	0.18	0.24
Joint	0.55	0.67	0.41	**0.48**	0.47	**0.64**	**0.52**	0.43	**0.63**	0.61	0.54
Proposed Method	**0.99**	**0.98**	**0.43**	0.44	**0.67**	0.46	0.25	**0.53**	0.61	**0.65**	**0.60**

Source: Developed by the author.

The evolution of the performance along the incremental learning process on the LIRIS dataset is presented in Figure 18. It can be noted that the proposed approach outperforms

the baseline methods in all incremental stages. The EWC and Rec-LwF approaches achieved comparable performance to the Finetune method.

Figure 18 – The average accuracy during the class-incremental learning training process on the LIRIS dataset



Source: Developed by the author.

The proposed method demonstrated a nearly-zero forgetting rate on the LIRIS dataset, as shown in Table 12. In comparison, the compared methods exhibited high forgetting rates. The Finetune and EWC methods achieved average forgetting rates of 0.39 and 0.38, respectively, while Rec-LwF achieved a forgetting rate of 0.27.

Table 12 – Forgetting measure on the LIRIS dataset.

Model/task	1	2	3	4	5	6	7	8	9	Avg
Finetune	0.70	0.83	0.25	0.19	0.43	0.38	0.27	0.28	0.21	0.39
EWC	0.70	0.78	0.25	0.17	0.40	0.37	0.22	0.28	0.22	0.38
Rec-LwF	0.79	0.79	0.16	0.07	0.08	0.28	0.25	0.02	0.01	0.27
Proposed Method	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.10**	**0.00**	**0.00**	**0.00**

Source: Developed by the author.

Table 13 presents the performance of the proposed method when evaluated on the MSR-VTT-subset dataset after learning all the tasks. The proposed model outperformed the compared methods in 75% of the tasks and achieved an overall score 8% higher than the Joint method, considered upper-bound. Also, the Joint method achieved the best accuracy in tasks 11, 16, 17, and 19, and the Finetune method achieved the best accuracy in task 20.

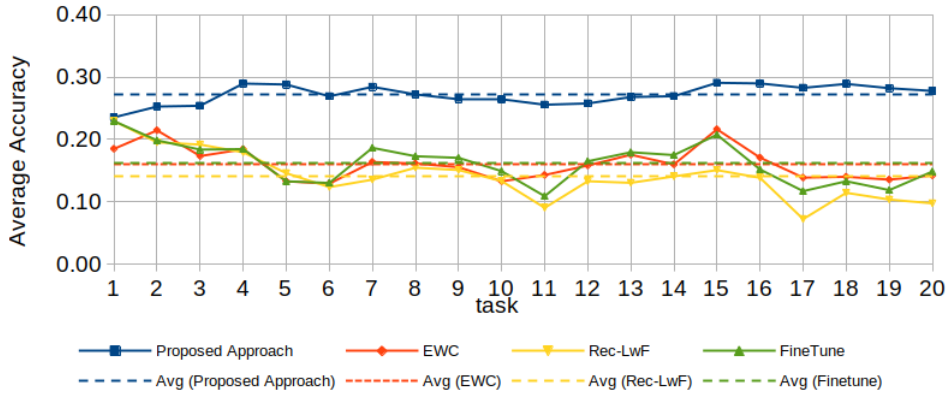
Table 13 – Average Accuracy after learning the last task on the MSR-VTT-subset dataset.

Model/task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Avg
Finetune	0.11	0.13	0.10	0.14	0.11	0.12	0.20	0.13	0.12	0.08	0.14	0.16	0.21	0.12	0.16	0.14	0.15	0.12	0.15	**0.38**	0.15
EWC	0.12	0.15	0.12	0.13	0.12	0.12	0.16	0.13	0.11	0.12	0.11	0.13	0.19	0.08	0.11	0.18	0.13	0.17	0.14	0.32	0.14
Rec-LwF	0.09	0.08	0.13	0.09	0.11	0.08	0.10	0.07	0.08	0.07	0.06	0.11	0.13	0.08	0.07	0.11	0.08	0.08	0.11	0.24	0.10
Joint	0.15	0.22	0.18	0.34	0.24	0.18	0.25	0.17	0.17	0.24	**0.28**	0.22	0.37	0.27	0.45	**0.24**	**0.28**	0.39	**0.28**	0.34	0.26
Proposed Method	**0.24**	**0.25**	**0.28**	**0.37**	**0.27**	**0.19**	**0.34**	**0.22**	**0.27**	**0.26**	0.24	**0.36**	**0.39**	**0.30**	**0.53**	0.15	0.16	**0.41**	0.15	0.19	**0.28**

Source: Developed by the author.

Figure 19 illustrates the evolution of the average accuracy along the incremental learning process on the MSR-VTT-subset dataset. Similarly, on the LIRIS dataset, the EWC approach achieved comparable performance to the FineTune method. In contrast, the Rec-LwF approach decreased performance with the growing number of tasks. This deterioration becomes more pronounced after task 10.

Figure 19 – The average accuracy during the class-incremental learning training process on the MSR-VTT-subset dataset



Source: Developed by the author.

In line with the findings reported in Table 12, the proposed method also achieved a nearly zero forgetting rate on MSR-VTT-subset dataset. The Finetune and EWC methods achieved average forgetting rates of 0,18 and 0,15, respectively, while Rec-LwF achieved a forgetting rate of 0,08.

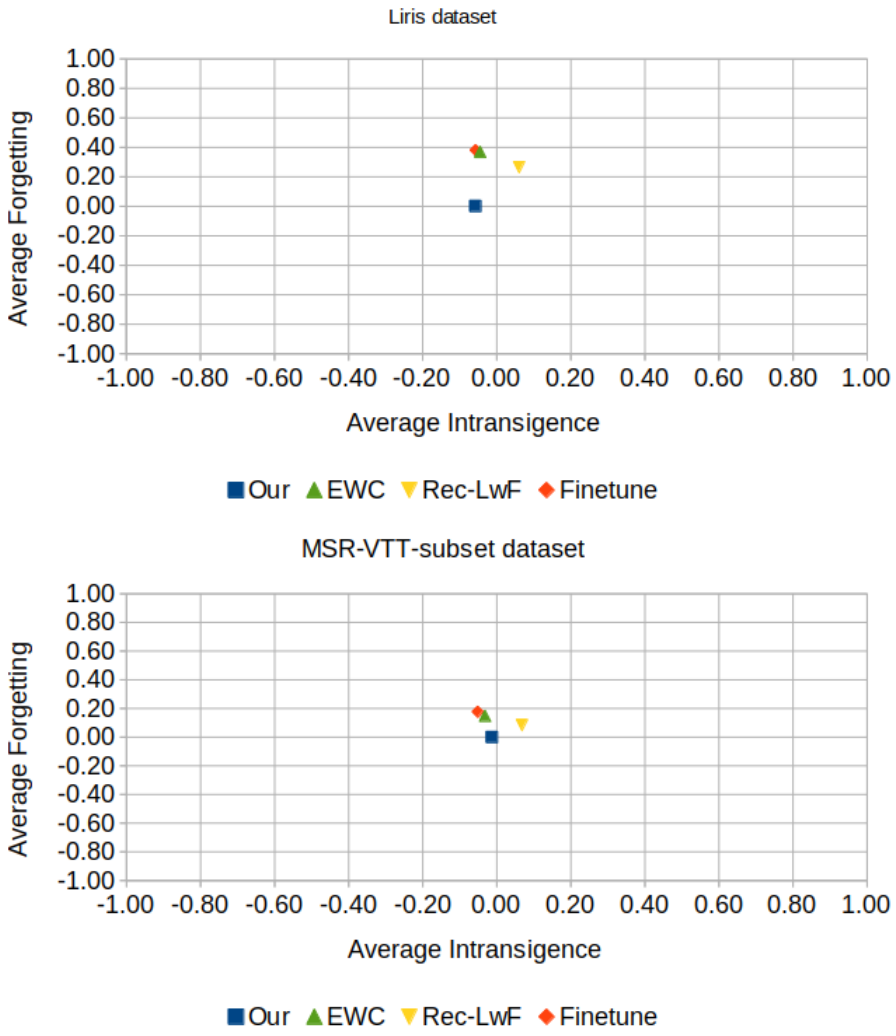
Table 14 – Forgetting measure on the MSR-VTT-subset dataset.

Model/task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	Avg
Finetune	0.10	0.13	0.14	0.19	0.15	0.05	0.16	0.04	0.13	0.15	0.12	0.21	0.22	0.20	0.53	0.18	0.11	0.31	0.21	0.18
EWC	0.06	0.16	0.12	0.22	0.11	0.02	0.16	0.08	0.12	0.15	0.12	0.13	0.15	0.08	0.52	0.17	0.07	0.18	0.17	0.15
Rec-LwF	0.11	0.11	0.13	0.22	0.08	0.02	0.07	0.05	0.06	0.07	0.06	0.05	0.07	0.06	0.08	0.09	0.09	0.06	0.10	0.08
Proposed Method	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**

Source: Developed by the author.

Figure 20 presents the interplay between Forgetting (stability) and Intransigence (Plasticity) measures, providing valuable insights into potential challenges related to the stability-plasticity dilemma. A model demonstrating low forgetting and low intransigence implies it can effectively learn new tasks while retaining previously acquired knowledge. Notably, the proposed approach demonstrated superior performance with nearly zero forgetting and negative intransigence in both datasets. The EWC and Finetune methods also achieved a negative intransigence score but high forgetting. In contrast, the LwF method showed high intransigence and forgetting.

Figure 20 – Interplay between Forgetting and Intransigence



Source: Developed by the author.

4.2.3.1 Discussion

The experimental results conducted on the LIRIS and MSR-VTT-subset datasets, depicted in Figures 18, 19, and 20, and presented in Tables 11 and 13 demonstrated that the proposed method consistently learned new tasks incrementally while maintaining performance in previously learned tasks. Furthermore, it outperformed the Joint method, which is considered an upper bound. This finding may indicate that the proposed model demonstrates superior capability in learning new tasks incrementally.

According to Santos *et al.* (2023), the combination of class imbalance and overlap features is one of the most challenging issues in machine learning. When training the model in a hold-out manner with combined classes, it needs to learn how to differentiate between classes and handle variations and overlaps in their feature spaces. Moreover, imbalanced datasets can

lead to biased learning, where the model becomes more proficient at predicting the dominant class but struggles with the minority class. The superior score achieved by the proposed method over the Joint method in both LIRIS and MSR-VTT datasets may indicate that it could deal adequately with such problems by focusing on learning task-specific patterns at each training step.

In Table 11, it is noticed that the proposed method achieves better accuracy (0.60) compared to other methods and even surpasses the Joint method on the LIRIS dataset, showcasing exceptional performance on certain tasks. Additionally, the proposed method outperforms other approaches in half of the tasks. In the MSR-VTT-subset dataset, it achieved better accuracy in 75% of the tasks. However, both the EWC and Rec-LwF approaches have slightly lower average performance compared to the lower bound model Finetune.

Catastrophic forgetting is a critical problem in class-incremental learning. As the model learns new tasks, there is a potential risk of either forgetting previously acquired knowledge or experiencing a significant decrease in performance. To address this challenge, the proposed method used attention-based masks as detailed in Section 3.2. These masks assign higher weights to important features or contextual information related to the current task, thereby preserving knowledge about previously learned tasks.

Although the Rec-LwF method achieved a similar average accuracy to the Finetune method in the LIRIS dataset, as presented in Table 11, it consistently outperformed across the tasks. The low average accuracy score can be attributed to the difficulty faced in learning the last task. Moreover, as presented in Table 12, it achieved a lower average forgetting rate of 0.27, in contrast to the Finetune method with a rate of 0.39. This finding also suggests that Rec-LwF tends to retain more information on average. The results further indicate that earlier tasks generally exhibit higher forgetting rates, emphasizing the challenge of preserving information from earlier stages.

The proposed method also outperforms the compared approaches in the MSR-VTT-subset dataset with an average accuracy of 0.28. This finding suggests that the proposed approach is also suitable for handling incremental learning tasks on large open datasets.

In terms of the forgetting rate, the proposed method achieved low forgetting rates for all tasks on both datasets. This observation suggests that the proposed model has the potential to mitigate the issue of catastrophic forgetting by retaining previously learned information when new tasks are introduced. Furthermore, this finding demonstrates the high stability of the

proposed method in class-incremental learning training. Rec-LwF also achieved lower forgetting rates compared to both EWC and Finetune methods, suggesting its superior ability to mitigate the issue of forgetting. However, it also achieved a low accuracy score, demonstrating difficulties in learning new classes incrementally.

The proposed method achieved a zero forgetting rate; nevertheless, experiments conducted on the MSR-VTT-subset dataset highlighted challenges in learning multiple tasks. Specifically, the proposed method faced difficulties in effectively learning new classes after task 15, as shown in Table 13. To retain knowledge from the previous tasks, it uses masks to constrain the network capacity. However, as the number of tasks increases, the capacity of the model to acquire new knowledge may become saturated. To address this issue, additional experiments were conducted to assess memory growth, and the findings are reported in Sections 4.2.6 and 4.2.7.

Notably, the accuracy scores achieved by the proposed method on the controlled LIRIS dataset surpass those achieved on the MSR-VTT-subset dataset. The variation in such scores can be explained by many distinctive factors. Firstly, the LIRIS dataset has a smaller number of videos and a more controlled vocabulary size, which limits the range of linguistic variations. Additionally, the environmental conditions under which the actions are performed in the LIRIS dataset demonstrate minimal variation, including only indoor scenes with consistent lighting conditions and the small number of individuals performing the actions in videos. In contrast, the MSR-VTT-subset dataset contains more complex videos and has a higher vocabulary diversity.

Regarding the trade-off between intransigence and forgetting, the proposed approach achieved the lowest score in both metrics, having the smallest distance from $(0,0)$. Such a finding may suggest it can adequately solve the class-incremental learning challenges. However, as discussed earlier, the capacity of the model to acquire new knowledge may reach saturation over time, and the essence of the intransigence score may not be fully captured. To investigate such a limitation, additional experiments will be presented and discussed in the next sections.

4.2.4 Experiment 2: Analysis of the Impact of the Initial Task Size on the Results

The experiments presented in this Section were motivated by the following research question: “Does the proposed method demonstrate superior performance when learning multiple classes in the initial task before gradually learning new classes, or does it achieve superior results by focusing on learning one class per task at a time?”.

In this experiment, the training process involved iterative steps for each dataset, with the proposed model being trained multiple times using different dataset splits. The overall accuracy was computed for each iteration, which was repeated a total of $n-1$ times, where n represents the total number of classes present in the dataset. Initially, the model was trained on individual tasks, with each task containing only one class. Subsequently, the initial task was gradually expanded to include additional classes in the following iterations. For instance, in the second trained model, the first task included the first two classes, while the remaining classes were learned incrementally, with one class per subsequent task. This incremental learning approach continued as subsequent models were trained, progressively incorporating an additional class into the initial task with each iteration until it incorporated $n-1$ classes. Finally, in the last stage, only one class remained to be learned incrementally. This sequential and progressive training methodology was implemented to evaluate whether the model achieves better performance when initially trained with many classes in the first task and subsequently learns fewer tasks incrementally or if better performance is achieved by initially training the first task with a small number of classes and gradually incorporating new classes.

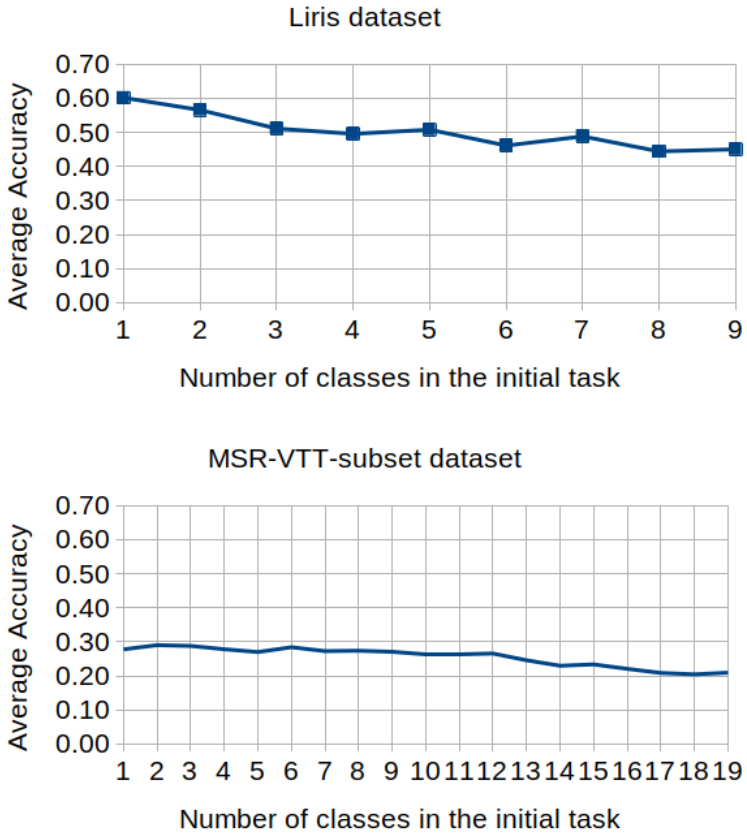
To assess the performance in these experiments, the Average Accuracy metric was used, as defined in Section 3.2.3. However, the accuracy measurement was replaced by the METEOR metric in our experiments, as it serves as a standard evaluation metric widely employed for video captioning assessment. As suggested by Masana *et al.* (2023), when dealing with tasks involving diverse numbers of classes, a class frequency weighted version is applied. Consequently, the average accuracy was calculated considering the number of classes in each task during the model evaluation.

Similar to Experiment 1, we set $k = 0$ in Algorithm 1. Thus, the model expansion process solely involves updating the input and output layers to incorporate new vocabulary.

Figure 21 illustrates the overall performance of the proposed model with respect to the number of classes in the first task. In the LIRIS dataset, the model exhibits a significantly higher performance of 0.60 when initially trained with a single class per task. However, the performance gradually decreases as the number of classes in the initial task increases. The performance ranges from 0.60 to 0.45 for 1 to 9 classes in the initial task. On the other hand, in the MSR-VTT-subset dataset, the model starts with a performance of 0.28 for a single class per task. The performance remains relatively stable for the first nine interactions, ranging between 0.27 and 0.29. However, the performance gradually declines from the 9th class onwards, reaching 0.21 for 19 classes in

the initial task.

Figure 21 – Average accuracy concerning the number of classes in the initial task.



Source: Developed by the author.

4.2.4.1 Discussion

The experimental results obtained from the LIRIS and MSR-VTT-subset datasets demonstrated that the proposed method performed better when trained with one class in the initial task before incrementally learning new tasks. This observation suggests that by training the model with individual classes per task, the proposed method can effectively focus on learning the distinctive characteristics of each class, leading to improved overall performance. On the other hand, when the proposed method was trained with multiple classes in the initial task, the average accuracy measure decreased. This finding also suggests that complexity and overlap among classes may pose challenges in distinguishing and generalizing patterns associated with each class. Moreover, more classes increase the risk of misclassification and confusion between similar classes, making the model learning process even more difficult. Overall, these findings highlight the importance of the proposed method in incrementally learning each class individually. By

focusing on one class at a time, the proposed model can mitigate the challenges of simultaneously learning multiple classes, thereby achieving better performance.

4.2.5 Experiment 3: Impact of Task Order on Overall Model Performance

The experiments presented in this Section were motivated by the following question: "Does the ordering of classes influence the overall performance of the proposed method?". The proposed method learns a sequence of tasks throughout the training process. These tasks can show significant differences in visual content or linguistic patterns employed to describe the scenes. Thus, six experiments were conducted to assess the influence of the task order on the performance of the proposed model. Similar to the previous experiments, it was set $k = 0$ in Algorithm 1 as the model expansion process in this experiment involves updating the input and output layers to incorporate new vocabulary.

First, the similarity between tasks was computed by measuring the distance between task centroids, treating each task as a cluster, and considering each reference sentence as data belonging to a cluster. Sentence-BERT (SBERT[6]) framework, which is based on the BERT model (DEVLIN *et al.*, 2019), was used to encode sentences into semantic embedding feature vectors (REIMERS; GUREVYCH, 2019). After encoding all the sentences, the centroid of each task was calculated by taking the mean of the features. The final similarity score was computed by employing the cosine similarity metric. Figure 22 presents the computed similarity matrices of classes for both LIRIS and MSR-VTT-subset datasets.

Based on the similarity matrices presented in Figure 22, six task sequences were created for each dataset. These sequences were created by considering the similarity, dissimilarity, and vocabulary size of the tasks. Considering the significant computational resources required to conduct experiments with all the different possible orders, two empirical sequences were defined for similar tasks (Sim1 and Sim2) and two for dissimilar tasks (Diss1 and Diss2). Note that the similarity and dissimilarity of the videos were computed based on the descriptions associated with each video task, and the sequences were defined based on the first task in the sequence. Also, each task within the sequences contains only one class, as presented in Sections 4.1.1.1 and 4.2.1.1. Furthermore, the task sequences, namely Descending Vocabulary Size (DVS) and Ascending Vocabulary Size (AVS) were defined, considering the vocabulary size of the tasks as outlined in Tables 3 and 10. The motivation behind the last two task sequences was to assess

[6] https://www.sbert.net/

Figure 22 – Similarity matrix based on reference sentence similarity.

Liris Dataset										
class	1	2	3	4	5	6	7	8	9	10
1	1.00	0.90	0.75	0.79	0.69	0.68	0.50	0.68	0.55	0.83
2	0.90	1.00	0.86	0.75	0.63	0.67	0.53	0.59	0.60	0.68
3	0.75	0.86	1.00	0.69	0.57	0.63	0.53	0.59	0.59	0.65
4	0.79	0.75	0.69	1.00	0.81	0.90	0.55	0.63	0.45	0.70
5	0.69	0.63	0.57	0.81	1.00	0.72	0.42	0.51	0.36	0.64
6	0.68	0.67	0.63	0.90	0.72	1.00	0.49	0.52	0.45	0.61
7	0.50	0.53	0.53	0.55	0.42	0.49	1.00	0.51	0.41	0.42
8	0.68	0.59	0.59	0.63	0.51	0.52	0.51	1.00	0.50	0.62
9	0.55	0.60	0.59	0.45	0.36	0.45	0.41	0.50	1.00	0.50
10	0.83	0.68	0.65	0.70	0.64	0.61	0.42	0.62	0.50	1.00

MSR-VTT subset dataset																				
class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1.00	0.56	0.69	0.68	0.72	0.74	0.75	0.76	0.70	0.68	0.76	0.68	0.72	0.92	0.88	0.70	0.92	0.53	0.61	0.76
2	0.56	1.00	0.82	0.56	0.63	0.80	0.55	0.63	0.66	0.71	0.54	0.53	0.53	0.51	0.53	0.55	0.50	0.58	0.47	0.47
3	0.69	0.82	1.00	0.71	0.68	0.83	0.66	0.80	0.74	0.80	0.65	0.68	0.64	0.59	0.66	0.66	0.59	0.65	0.58	0.56
4	0.68	0.56	0.71	1.00	0.66	0.75	0.65	0.81	0.72	0.68	0.57	0.90	0.61	0.61	0.66	0.77	0.62	0.66	0.55	0.55
5	0.72	0.63	0.68	0.66	1.00	0.82	0.89	0.77	0.85	0.77	0.74	0.70	0.86	0.71	0.74	0.74	0.71	0.66	0.67	0.68
6	0.74	0.80	0.83	0.75	0.82	1.00	0.73	0.85	0.86	0.83	0.67	0.73	0.70	0.68	0.71	0.74	0.69	0.69	0.61	0.64
7	0.75	0.55	0.66	0.65	0.89	0.73	1.00	0.74	0.78	0.69	0.74	0.66	0.96	0.74	0.78	0.66	0.72	0.58	0.60	0.63
8	0.76	0.63	0.80	0.81	0.77	0.85	0.74	1.00	0.85	0.80	0.72	0.78	0.70	0.70	0.77	0.76	0.71	0.68	0.62	0.67
9	0.70	0.66	0.74	0.72	0.85	0.86	0.78	0.85	1.00	0.86	0.69	0.71	0.76	0.65	0.69	0.74	0.64	0.80	0.69	0.61
10	0.68	0.71	0.80	0.68	0.77	0.83	0.69	0.80	0.86	1.00	0.65	0.66	0.68	0.62	0.65	0.69	0.61	0.73	0.60	0.59
11	0.76	0.54	0.65	0.57	0.74	0.67	0.74	0.72	0.69	0.65	1.00	0.58	0.72	0.73	0.91	0.61	0.73	0.53	0.56	0.65
12	0.68	0.53	0.68	0.90	0.70	0.73	0.66	0.78	0.71	0.66	0.58	1.00	0.61	0.61	0.64	0.87	0.62	0.66	0.62	0.59
13	0.72	0.53	0.64	0.61	0.86	0.70	0.96	0.70	0.76	0.68	0.72	0.61	1.00	0.73	0.76	0.62	0.69	0.61	0.61	0.58
14	0.92	0.51	0.59	0.61	0.71	0.68	0.74	0.70	0.65	0.62	0.73	0.61	0.73	1.00	0.86	0.65	0.94	0.48	0.62	0.77
15	0.88	0.53	0.66	0.66	0.74	0.71	0.78	0.77	0.69	0.65	0.91	0.64	0.76	0.86	1.00	0.66	0.87	0.53	0.57	0.72
16	0.70	0.55	0.66	0.77	0.74	0.74	0.66	0.76	0.74	0.69	0.61	0.87	0.62	0.65	0.66	1.00	0.66	0.64	0.66	0.66
17	0.92	0.50	0.59	0.62	0.71	0.69	0.72	0.71	0.64	0.61	0.73	0.62	0.69	0.94	0.87	0.66	1.00	0.47	0.57	0.79
18	0.53	0.58	0.65	0.66	0.66	0.69	0.58	0.68	0.80	0.73	0.53	0.66	0.61	0.48	0.53	0.64	0.47	1.00	0.63	0.44
19	0.61	0.47	0.58	0.55	0.67	0.61	0.60	0.62	0.69	0.60	0.56	0.62	0.61	0.62	0.57	0.66	0.57	0.63	1.00	0.60
20	0.76	0.47	0.56	0.55	0.68	0.64	0.63	0.67	0.61	0.59	0.65	0.59	0.58	0.77	0.72	0.66	0.79	0.44	0.60	1.00

Source: Developed by the author.

the performance of the proposed method in two distinct scenarios: one where the initial tasks contain a wide-ranging and diverse vocabulary and the other where the model is fed with only a limited set of words during each task.

Table 15 presents the experimental results obtained from the LIRIS dataset. It can be noted that the ordering of classes does influence the performance of the proposed method. The AVS and Diss2 sequences reached the highest average accuracy of 0.77, followed by Sim1 order with a score of 0.72, Sim2 with a score of 0.67, and Diss1 with a score of 0.63. On the other hand, the DVS order achieved the lowest average accuracy of 0.59.

When analyzing the scores of the tasks across different sequences, it is noteworthy that specific tasks (such as T3, T4, and T8) achieved a consistent score with minimal variation throughout the sequences. Conversely, tasks T2, T5, T6, T9, and T10 displayed substantial score variations across different task sequences. Additionally, when the T7 task was trained at the end of the sequence, there was a significant decrease in the average accuracy score.

Table 15 – Average accuracy on the LIRIS dataset in different task sequences. M denotes the METEOR score achieved after training the last task.

Sim1	M	Sim2	M	Diss1	M	Diss2	M	DVS	M	AVS	M
T1	0.99	T4	0.46	T7	0.47	T1	0.99	T3	0.42	T7	0.47
T2	0.98	T6	0.54	T9	1.00	T7	0.47	T1	0.99	T9	1.00
T10	0.96	T5	0.96	T8	0.53	T5	0.96	T6	0.53	T10	0.65
T4	0.51	T1	0.98	T4	0.49	T8	0.52	T4	0.45	T2	0.98
T3	0.42	T2	0.98	T6	0.55	T2	0.94	T8	0.54	T5	0.96
T5	0.96	T10	0.53	T5	0.92	T9	1.00	T5	0.59	T8	0.70
T6	0.59	T3	0.42	T1	0.72	T10	0.96	T2	0.98	T4	0.59
T8	0.52	T8	0.55	T2	0.66	T6	0.94	T10	0.53	T6	0.94
T9	1.00	T7	0.60	T10	0.53	T3	0.40	T9	0.60	T1	0.99
T7	**0.29**	T9	0.66	T3	0.39	T4	0.53	T7	**0.22**	T3	0.40
avg	0.72	avg	0.67	avg	0.63	avg	0.77	avg	0.59	avg	0.77

Source: Developed by the author.

Table 16 presents the experimental results conducted on the MSR-VTT-subset dataset for the six distinct sequences. It is worth highlighting that the scores obtained in this dataset are considerably lower in comparison to those presented in Table 15. This disparity can be attributed to the complexity and diversity of video content and the notably more extensive vocabulary contained within the MSR-VTT-subset dataset, resulting in a more challenging dataset in contrast to the LIRIS dataset.

Table 16 – Average accuracy on the MSR-VTT-subset dataset in different task sequences. M denotes the METEOR score achieved after training the last task.

Sim1	M	Sim2	M	Diss1	M	Diss2	M	DVS	M	AVS	M
T8	0.22	T1	0.24	T19	0.38	T18	0.42	T5	0.28	T19	0.38
T6	0.15	T14	0.26	T2	0.26	T5	0.27	T2	0.27	T13	0.40
T9	0.29	T17	0.28	T17	0.25	T2	0.26	T4	0.38	T14	0.25
T10	0.31	T20	0.39	T18	0.44	T4	0.38	T10	0.28	T18	0.42
T5	0.26	T15	0.64	T1	0.19	T15	0.66	T15	0.58	T3	0.25
T7	0.38	T11	0.26	T4	0.38	T8	0.22	T17	0.25	T16	0.26
T13	0.40	T5	0.27	T13	0.37	T7	0.36	T12	0.35	T11	0.16
T15	0.60	T7	0.35	T20	0.44	T16	0.39	T1	0.20	T9	0.15
T11	0.26	T13	0.41	T12	0.27	T3	0.25	T20	0.41	T7	0.36
T1	0.21	T9	0.25	T14	0.25	T13	0.43	T8	0.22	T6	0.14
T14	0.22	T10	0.24	T3	0.15	T12	0.32	T6	0.20	T8	0.15
T17	0.30	T6	0.17	T11	0.16	T17	0.24	T7	0.36	T20	0.43
T20	0.43	T3	0.14	T16	0.16	T9	0.23	T9	0.24	T1	0.14
T16	0.17	T2	0.27	T7	0.36	T20	0.20	T11	0.16	T12	0.17
T12	0.18	T8	0.15	T10	0.24	T6	0.17	T16	0.25	T17	0.17
T4	0.38	T4	0.27	T15	0.56	T19	0.19	T3	0.17	T15	0.37
T3	0.16	T12	0.15	T9	0.19	T14	0.22	T18	0.20	T10	0.19
T2	0.14	T16	0.16	T8	0.15	T11	0.15	T14	0.24	T4	0.17
T19	0.18	T19	0.20	T5	0.14	T10	0.19	T13	0.28	T2	0.17
T18	0.27	T18	0.31	T6	0.15	T1	0.16	T19	0.18	T5	0.15
Avg	0.28	Avg	0.27	Avg	0.28	Avg	0.29	Avg	0.27	Avg	0.24

Source: Developed by the author.

The overall average accuracy presented in Table 16 ranges from 0.24 to 0.29, indicating

a consistent and minimal discrepancy between them. However, it is worth noting that the scores of the tasks show variations based on their respective order within the training sequence. For instance, tasks T5, T13, T18, and T19 demonstrated higher performance scores when trained at the initial positions within the task sequence, but their performance notably declined when trained at the latter stages of the sequence. It can also be noted that the performance of the proposed method decreases after task 10 within the AVS sequence. It can also be noted in Table 16 that the T6 task achieved consistently low scores across various task sequences, even when trained within the initial tasks. On the other hand, the T15 task achieved a high score even when trained at the end of the sequence.

4.2.5.1 Discussion

According to the experimental results, the average accuracy varied according to the task orders. In the LIRIS dataset, results show the score achieved by the proposed method ranged from 0.59 to 0.77, demonstrating that the order of the tasks affects the overall score on such dataset.

Moreover, it was noted that the similarity or dissimilarity between classes could not be considered the only aspect that influences the performance of the model. Such findings suggest that other aspects beyond similarity or dissimilarity between tasks also influence the performance of the proposed model.

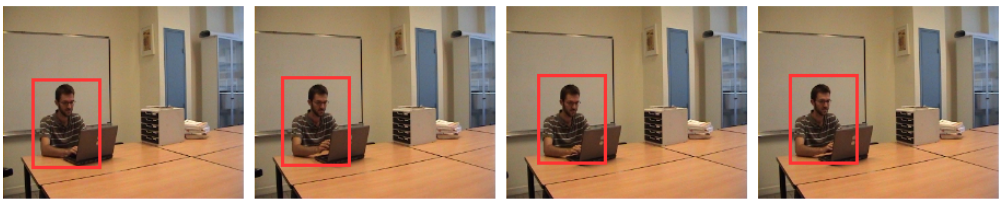
To better understand the score variations achieved by the proposed approach across different task sequences, a qualitative analysis was also conducted. Figure 23 depicts the captions generated by the proposed method for three distinct videos obtained from the LIRIS dataset across task sequences with the reference sentences provided by the dataset for each video. Notably, the proposed method generated some captions that were identical to those presented in the reference sentence list.

Figure 23 – Qualitative analysis of generated descriptions on the LIRIS dataset int different task sequences.

A) **Task**: T7 **Video**: vid0137_7_38_82

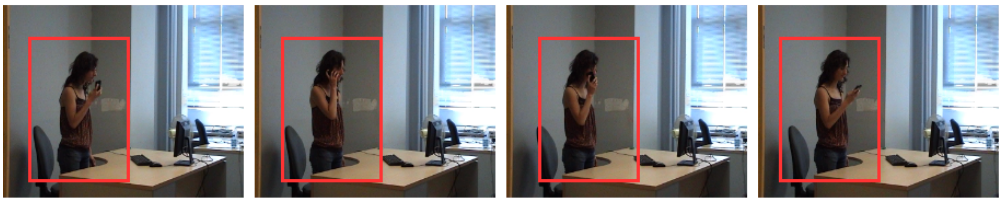
Sequence	Predicted Sentence	Reference Sentences
Sim1	a man **is an an office**	a person leaves baggage unattended
Sim2	a man is leaving baggage unattended	a man leaves baggage unattended
Diss1	a man in a **black shirt** leaves baggage unattended	a person is leaving baggage unattended
Diss2	a man in a **black shirt** leaves baggage unattended	a man is leaving baggage unattended
DVS	a man is leaving **an office**	a man in a blue t-shirt leaves baggage unattended
AVS	a man in a **black shirt** leaves baggage unattended	

B) **Task**: T9 **Video**: vid0041_9_1_90



Sequence	Predicted Sentence	Reference Sentences
Sim1	a person is typing on a keyboard	a person types on a keyboard
Sim2	a person is typing on a keyboard	a man types on a keyboard
Diss1	a man is typing on a keyboard	a person is typing on a keyboard
Diss2	a man is typing on a keyboard	a man is typing on a keyboard
DVS	a man is **putting** an a keyboard	a man in a striped t-shirt types on a keyboard
AVS	a man is typing on a keyboard	

C) **Task:** T10 **Video:** vid0027_10_153_291



Sequence	Predicted Sentence	Reference Sentences
Sim1	a **man** is talking on a telephone	a person talks on a telephone
Sim2	a **man** in a black **t-shirt** talks on a telephone	a woman talks on a telephone
Diss1	a **man** in a black **t-shirt** talks on a telephone	a person is talking on a telephone
Diss2	a **man** is talking on a telephone	a woman is talking on a telephone
DVS	a **man** in a black **t-shirt** talks on a telephone	a woman in a black blouse talks on a telephone
AVS	a person is on a telephone	

Source: Developed by the author.

Figure 23A shows a randomly selected video from the T7 task. It can be observed that

the proposed method predicted incorrect sentences for the analyzed video within the Sim1 and DVS task orders. Task T7, positioned at the end of such sequences, notably achieved low scores of 0.29 and 0.22, respectively. These results indicate that the model faced difficulties in effective learning and describing the videos associated with this task in these specific task orders. Moreover, the generated descriptions for this video in the remaining task sequences correctly described the action but included incorrect details regarding soft biometrics, specifically clothing color. Similar findings can be observed in Figure 23C, further highlighting the difficulty of the model in learning and describing this particular aspect accurately. To address this limitation, potential strategies could involve incorporating more specific features into the model or considering a post-processing step to incorporate the soft biometrics information into the predicted sentence.

Figure 23B shows a randomly selected video presented in Task T9. It can be noted that the proposed method could generate captions identical to one of the reference sentences in almost all task orders. However, for the DVS order, which achieved a score of 0.59, as presented in Table 15, the model predicted a sentence with an incorrect action for the given video. While an accurate description was generated in the Sim2 order, the score obtained was 0.66, and both of these tasks were learned at the end of the task sequence. This observation may suggest that the proposed model is more prone to learn new patterns and vocabulary from novel tasks when they are learned at the beginning of the sequence. Nevertheless, surprisingly, the T9 task achieved its highest score in order Sim1 when trained near the end of the sequence. This outcome indicates that the proposed model faces additional challenges beyond task order to efficiently learn new tasks, particularly when confronted with a substantial number of sequential tasks.

In Figure 23C), it can be noted that the proposed method consistently demonstrated proficiency in accurately describing the action depicted in the video. However, it struggled to provide precise descriptions of the soft biometrics information, specifically about gender and clothing details. Notably, within the Sim1 and Diss2 sequences, where clothing information was not described, the corresponding scores obtained (as indicated in Table 15) were comparatively higher than those obtained in sequences where clothing descriptions were inaccurately provided. This observation demonstrates the sensitivity of the metric when evaluating short sentences, wherein slight variations of a few words significantly impact the computed score. Moreover, this finding aligns with the findings reported by (INÁCIO, 2023), highlighting the limitations of widely-used metrics in video captioning evaluation concerning their ability to assess the semantic aspects of sentences accurately.

Similar to the findings from the LIRIS dataset, the results obtained from the MSR-VTT-subset dataset, as presented in Table 16, show that certain tasks, specifically T5, T13, T18, and T19, achieved higher scores when trained at the beginning of the task sequence. However, their performance notably declined when trained toward the final of the sequence.

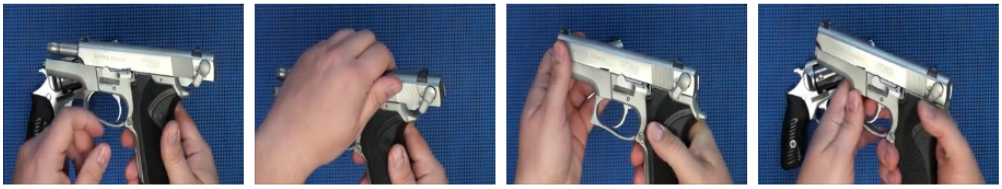
This empirical observation suggests that the proposed method may face challenges in learning many sequential tasks. It can also be noted that the performance of the proposed method decreases after 10-th task within the AVS sequence. This observation may raise a limitation capacity of the model to effectively acquire new knowledge when confronted with tasks that involve a substantial amount of new vocabulary.

The use of attention masks may contribute to this capacity limitation. As the proposed model learns new tasks, the masks retain information regarding the importance of neurons in describing previous tasks. Consequently, when exposed to new tasks characterized by considerable new vocabularies, the available memory for learning new knowledge may be constrained. Such a limitation is explored in the following experiments.

In Table 16, it can also be noted that the T6 task achieved consistently low scores across various task sequences, even when trained within the initial tasks. This may suggest that the proposed method could not properly learn such a task. On the other hand, the T15 task achieves a high score even when trained at the end of the sequence, indicating that it was an easily learnable task and demanded minimal computation effort from the model.

Figure 24 – Qualitative analysis of generated descriptions on the MSR-VTT-subset dataset in different task sequences.

A) **Task:** T8 **Vídeo:** 504



Sequence	Predicted Sentence	Reference Sentences
Sim1	a man is talking about a gun	a guy talking about his gun
Sim2	a man is talking about a man	a man is explaining on how to use a gun
Diss1	**a man is a man**	a man showing a little bit of how a gun works
Diss2	a man is showing a gun and aiming it	someone is showing a gun
DVS	a man is talking about a gun	a man is talking about a pistol
AVS	**a man is a a**	a man reviews a firearm and explains recoil and jamming

B) **Task**: T2 **Vídeo**: 1111

Sequence	Predicted Sentence	Reference Sentences
Sim1	**a man is talking to a woman in a car**	a cook is preparing a salad
Sim2	a man is cooking in a kitchen	a man adds greens to boiling water
Diss1	a man is cooking a dish in a kitchen	a man cooking food
Diss2	a man is cooking in a kitchen	a man is cooking in a kitchen
DVS	a man is cooking	a man is talking while is cooking explaining how to make a dish
AVS	**a man is a woman is a**	person is preparing some food

C) **Task**: T18 **Vídeo**: 3856

Sequence	Predicted Sentence	Reference Sentences
Sim1	**a man is playing a a video**	a person is showing a stroller
Sim2	**a woman is talking about a video**	lady showing how to operate stroller
Diss1	a woman is talking about a baby stroller	a woman is giving demo for baby trolley
Diss2	a woman is showing a stroller	someone is talking about different types of baby strollers
DVS	**a woman is dancing**	strollers on the side of a house
AVS	a woman is showing a stroller	a woman talks about how to use a stroller

Source: Developed by the author.

A qualitative analysis was also performed based on the results obtained from the MSR-VTT-subset to examine the task score variations in different sequences. Figure 24 presents a qualitative analysis of the descriptions generated by the proposed method for three selected videos obtained from the MSR-VTT-subset dataset across different task sequences.

In Figure 24A, it is noticed that the proposed method did not generate a fluent sentence (highlighted in red) for the analyzed videos in the sequences Diss1 and AVS. For both task sequences, the model learned them in the final sequence, confirming that the proposed method may not deal with many tasks sequentially, probably due to memory limitation. The same finding can be seen in AVS sequence of Task T2 in Figure 24B.

Figure 24B and 24C present sentences highlighted in red that, although fluent, do not accurately describe the corresponding videos. These sentences belong to sequence tasks learned in the final sequence, demonstrating that the proposed method may face challenges in learning many new tasks, potentially due to the limited memory capacity discussed previously. By preserving the knowledge learned in previous tasks, the method emphasizes stability rather than plasticity, suggesting it may face a challenge in effectively balancing the stability-plasticity dilemma.

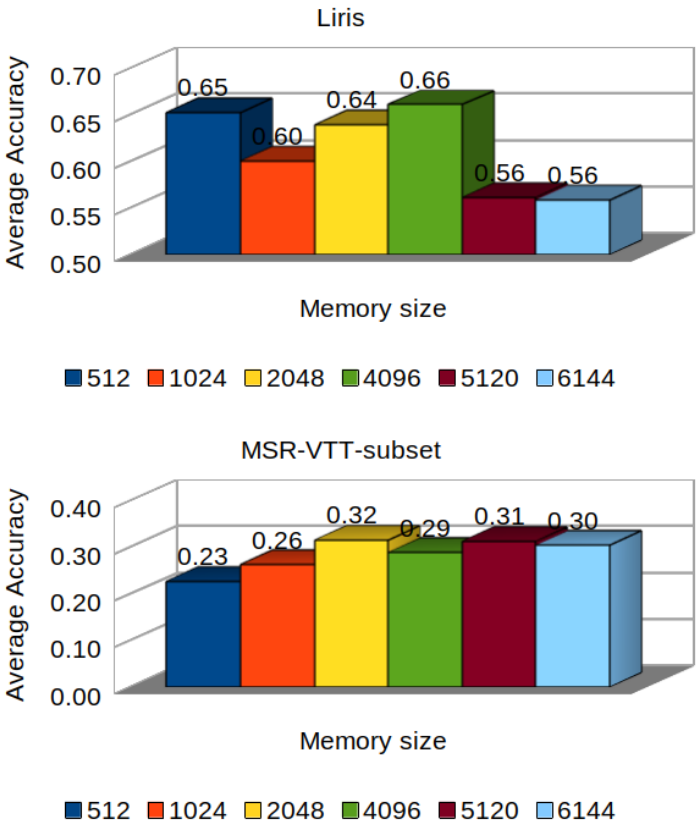
Based on the observed findings in this experiment regarding the potential limitations of memory in the incremental learning of new tasks, the experiments presented in the following sections were conducted to address this issue.

4.2.6 Experiment 4: Initial memory size

The experiments presented in this section were motivated by the following question: “How does the size of the internal memory affect the overall performance of the proposed method?”. Based on the experimental results presented in Section 4.2.5, a potential limitation of the proposed method has been identified concerning its memory size, as the method showed difficulty in learning many new tasks sequentially.

Experiments were performed with six different memory sizes: 512, 1024, 2048, 4096, 5120, and 6144. Figure 25 presents the average accuracy on both LIRIS and MSR-VTT-dataset dataset.

Figure 25 – Average accuracy across various memory sizes.



Source: Developed by the author.

For the LIRIS dataset, the model exhibits relatively stable performance within the memory size range from 512 to 2048. The highest average accuracy (0.66) is achieved with a memory size of 4096, indicating that this dataset does not necessitate higher memory capacity to learn the patterns adequately. Conversely, the MSR-VTT-subset dataset achieved lower scores with memory sizes of 512 and 1024 but demonstrated stable performance from 2048 to 6144. Its peak average accuracy (0.32) is achieved with a memory size of 2048, signifying that this dataset requires greater memory capacity to learn the patterns effectively. This may suggest that the optimal memory size may vary depending on the dataset and the specific tasks being performed. Furthermore, both datasets demonstrate varying sensitivity to changes in memory size, with some memory capacities leading to significant fluctuations in performance scores. However, both datasets show relatively consistent performance patterns across memory sizes, indicating robustness to some extent.

4.2.6.1 Discussion

The memory capacity of a machine learning model plays a key role in its learning capabilities, generalization performance, and predictive abilities. The experimental results presented in Figure 25 show that the proposed model achieved superior performance on the LIRIS dataset when using a memory size 4096. In contrast, the performance declined when training the model with larger memory sizes. This observation highlights the potential risks associated with employing a large memory model with a small dataset, namely the phenomenon of overfitting. Overfitting occurs when a model becomes excessively specialized to the training data, resulting in reduced performance on unseen data and hindered generalization capabilities. Employing a large memory in machine learning models introduces additional complexity, primarily in the form of increased parameters. Consequently, training such a model with limited data can pose challenges, as the high capacity of the model may only be effectively beneficial with a sufficient amount of data to capture meaningful patterns and relationships.

On the other hand, the proposed method achieved better performance on the MSR-VTT-subset dataset when it was trained with a larger memory size (2048). As the MSR-VTT-subset dataset is larger and more diverse in video content and vocabulary compared to the LIRIS dataset, a larger memory capacity becomes necessary to achieve superior performance. While a larger memory capacity can offer advantages in such a dataset, some concerns should be observed regarding model architecture, optimization techniques, and high demand for computational resources.

Finding a balance between model complexity and available resources is crucial for achieving optimal performance scores during machine learning model training. The experiments conducted in this Section provide evidence that internal memory size can significantly impact overall performance. Determining the appropriate memory size poses a challenge, as it requires careful consideration of the size and complexity of the dataset being used.

4.2.7 Experiment 5: Dynamic memory expansion

These experiments aim to tackle the limitation pointed out in the previous experiments regarding the possible difficulty of the proposed method in learning many new tasks incrementally. The question that motivated the experiments in this section is: “Does increasing the memory of the proposed method improve its capacity to learn new tasks? How and at what stages of

the training process should the memory expansion be implemented to optimize the learning performance?". The underlying premise is that the model requires additional memory capacity to acquire new task-specific knowledge when faced with a new task. Thus, the memory of the proposed model can be expanded by a size of s, as proposed in Algorithm 1, before starting to learn a new task.

The experiments were conducted using four distinct settings: two settings employing linearly increasing memory with $s = 10$ and $s = 50$ before training a new task $t > 1$, and two settings employing dynamic memory expansion with $s = 10$ and $s = 50$. It is important to note that, in the experiments involving linearly increasing memory, the threshold k was set to 1 to ensure consistent memory expansion before training a new task, regardless of the model's capacity to acquire new knowledge.

Table 17 presents the average accuracy and average forgetting of the proposed method in five different memory expansion settings for various initial memory sizes, using the LIRIS and MSR-VTT-subset dataset. Notably, significant improvements in the performance of the proposed method were observed in comparison to the "no memory expansion" scenario, even with a slight increase in the forgetting rate resulting from the memory expansion process.

In the LIRIS dataset, the highest score achieved was 0.76, using an initial memory size of 512 and employing dynamic memory expansion by 10, following the process described in the Algorithm 1. Conversely, for the MSR-VTT-subset dataset, best results were achieved with an initial memory size of 2048 and employing memory expansion either linearly by 10 or dynamically by 10 or 50. While comparable results were observed with initial memory sizes of 5120 and 6144, a slightly higher standard deviation was observed in those scores, indicating more variation in the individual performance of tasks in those scenarios.

4.2.7.1 Discussion

The experimental results show that memory expansion significantly enhanced the performance of the proposed approach. In particular, the model trained with an initial memory size of 512, combined with dynamic memory expansion increments of 10, achieved the highest overall score on the LIRIS dataset. These findings indicate that in a small and controlled dataset, such as the LIRIS dataset, the proposed model achieved lower computational costs by initially starting with a smaller memory size, such as 512, and expanding it on demand.

Regarding the model with an initial memory size of 4096 trained on the LIRIS dataset,

Table 17 – Average accuracy and Average forgetting, with Standard Deviation, in five different memory expansion settings in different initial memory sizes.

Initial Memory	Setting	Liris		MSR-VTT-subset	
		Avg. Acc.	Forgetting	Avg. Acc.	Forgetting
512	No memory expansion	0.65 ± 0.33	0.00 ± 0.00	0.23 ± 0.08	0.00± 0.00
	Linear expansion by 10	0.68 ± 0.23	0.07 ± 0.17	0.28 ± 0.07	0.02 ± 0.02
	Linear expansion by 50	0.73 ± 0.24	0.11 ± 0.18	**0.30 ± 0.06**	0.01 ± 0.02
	Dynamic expansion by 10	**0.76 ± 0.23**	0.01 ± 0.02	0.24 ± 0.08	0.05 ± 0.06
	Dynamic expansion by 50	0.68 ± 0.23	0.09 ± 0.14	0.30 ± 0.07	0.02 ± 0.02
1024	No memory expansion	0.60 ± 0.24	0.00 ± 0.00	0.26 ± 0.07	0.00 ± 0.00
	Linear expansion by 10	**0.69 ± 0.25**	0.00 ± 0.04	0.31 ± 0.07	0.02 ± 0.03
	Linear expansion by 50	0.60 ± 0.20	0.10 ± 0.19	0.30 ± 0.07	0.02 ± 0.02
	Dynamic expansion by 10	0.66 ± 0.24	0.04 ± 0.10	**0.31 ± 0.05**	0.02 ± 0.02
	Dynamic expansion by 50	0.67 ± 0.24	0.03 ± 0.10	0.30 ± 0.07	0.03 ± 0.04
2048	No memory expansion	0.64 ± 0.25	0.00 ± 0.00	0.32 ± 0.07	0.00 ± 0.00
	Linear expansion by 10	0.65 ± 0.23	0.00 ± 0.00	**0.32 ± 0.06**	0.02 ± 0.02
	Linear expansion by 50	0.60 ± 0.21	0.00 ± 0.01	0.30 ± 0.06	0.03 ± 0.04
	Dynamic expansion by 10	**0.67 ± 0.23**	0.01 ± 0.03	**0.32 ± 0.06**	0.01 ± 0.01
	Dynamic expansion by 50	0.65 ± 0.24	0.00 ± 0.01	**0.32 ± 0.06**	0.01 ± 0.02
4096	No memory expansion	**0.66 ± 0.23**	0.00 ± 0.00	0.29 ± 0.07	0.00 ± 0.00
	Linear expansion by 10	0.61 ± 0.21	0.00 ± 0.00	**0.31 ± 0.07**	0.01 ± 0.01
	Linear expansion by 50	0.59 ± 0.21	-0.01 ± 0.02	**0.31 ± 0.07**	0.01 ± 0.02
	Dynamic expansion by 10	0.62 ± 0.20	0.00 ± 0.00	0.31 ± 0.08	0.02 ± 0.03
	Dynamic expansion by 50	0.61 ± 0.21	0.00 ± 0.00	0.31 ± 0.08	0.03 ± 0.03
5120	No memory expansion	0.56 ± 0.15	0.00 ± 0.00	0.31 ± 0.07	0.00 ± 0.00
	Linear expansion by 10	0.55 ± 0.16	0.00 ± 0.01	0.30 ± 0.07	0.01 ± 0.02
	Linear expansion by 50	0.57 ± 0.16	0.00 ± 0.01	0.31 ± 0.07	0.01 ± 0.02
	Dynamic expansion by 10	0.58 ± 0.16	0.00 ± 0.01	**0.32 ± 0.07**	0.01 ± 0.02
	Dynamic expansion by 50	**0.63 ± 0.21**	-0.01 ± 0.02	0.30 ± 0.06	0.02 ± 0.03
6144	No memory expansion	0.56 ± 0.17	0.00 ± 0.00	0.30 ± 0.07	0.00 ± 0.00
	Linear expansion by 10	0.56 ± 0.16	0.00 ± 0.01	0.30 ± 0.07	0.02 ± 0.01
	Linear expansion by 50	**0.60 ± 0.21**	0.00 ± 0.01	0.30 ± 0.06	0.02 ± 0.02
	Dynamic expansion by 10	0.51 ± 0.06	0.01 ± 0.01	**0.32 ± 0.07**	0.01 ± 0.02
	Dynamic expansion by 50	0.52 ± 0.08	0.00 ± 0.02	0.30 ± 0.07	0.02 ± 0.02

Source: Developed by the author.

the memory expansion strategy did not improve the performance.

Regarding the models with initial sizes of 4096, 5120, and 6144, the memory growth models obtained marginally better results than those obtained with fixed memory size. Such findings indicate that the increased model complexity, introduced by larger memory sizes, may have a limited impact on learning a small dataset. However, further analysis and experimentation are necessary to fully understand the relationship between memory size, model complexity, and dataset size.

The proposed memory expansion method also notably enhanced the performance of the proposed method when trained on the MSR-VTT-subset dataset with small initial memory sizes, such as 512 or 1024. However, as the initial memory size exceeded 1024, the improvement became less significant. This observation may suggest that when training the proposed method

using a larger and more complex dataset, defining a higher initial memory size may be necessary to learn novel tasks effectively.

Regarding the average rate of forgetting, it can be observed that memory expansion led to an increase in the forgetting rate, mainly in the case of trained models with lower initial memory sizes (512 or 1024). Despite this increase, the improvement achieved in the overall performance outweighed the impact of heightened forgetting rates. This finding indicates that the advantages of memory expansion in improving the model's capacity to learn new tasks outweigh the potential drawback of increased forgetting, especially for models trained with limited initial memory.

According to the experimental results, no clear relationship was observed between the initial memory size and the memory expansion method (linear or dynamic), nor the extent of expansion on both datasets. However, it is worth noting that higher scores in the LIRIS dataset were achieved with a smaller initial memory. In contrast, in the MSR-VTT-subset dataset, the proposed method achieved its best performance when trained with a larger initial memory. This observation further supports the result obtained in Section 4.2.6, which indicates that a large dataset with diverse vocabulary requires a higher memory capacity to enhance pattern learning.

5 CONCLUSIONS AND FUTURE WORK

It is a matter of fact that most artificial intelligence methods rely on the closed-set assumption. The same also holds for the specific case of automatic video description systems. Existing methods based on a closed-set perspective can describe only the temporal events previously seen during the training step. Unless one trains models using datasets that contain all the existing events and actions of interest, their ability to recognize unknown events within videos in uncontrolled environments will be limited. Furthermore, it will be difficult for models to learn and adapt to these unknown events after training. In addition, most current video description approaches focus only on single actions co-occurring, while concurrent events may occur in the real world.

In Chapter 1, the research questions were introduced, namely, "Can a video description framework be designed to describe in natural language concurrent known video events in different contexts and deal with the unknown ones? What strategies could be employed to enhance the framework's ability to learn and describe unknown events?" Along with the development of this thesis, we investigated and proposed novel methods to detect and describe concurrent events in an open-set scenario (Section 3.1), as well as a video captioning approach in a class-incremental learning scenario (Section 3.2). The experiments of Chapter 4 showed promising insights. The findings from these experiments indicate that developing a deep learning-based approach to address the video description problem in an open-world environment is feasible, even considering the potential emergence of new actions and vocabularies beyond the initially trained model. The results obtained demonstrate the effectiveness of the model for dealing with unknown events. Additionally, the proposed incremental learning method for video captioning provides a perspective on how a deep learning model can systematically acquire the capability to describe incrementally new and unknown events over time.

The general objective of this thesis, referred to as "To propose methods for the semantic description of videos in an open world scenario," has been accomplished through the development of the methods proposed in Chapter 3. The OSVidCap (INÁCIO *et al.*, 2021) framework presented in Section 3.1 describes both simple and concurrent events, including detecting unknown events. Such a framework can easily be extended to incorporate different features from different modalities, including audio and features representing the relationship between objects. The class incremental learning method presented in Section 3.2 allows the model to learn

new knowledge incrementally without requiring complete retraining. Although the incremental learning method proposed employs a similar encoder/decoder structure as OSVidCap, it cannot be asserted that it is applicable in an open-world scenario. Such a limitation arises from the dependence of the Open-set module on an EVM, making it impossible to adapt to the context of incremental learning. Addressing this challenge would require enhancing the open-set module introduced in OSVidCap with a mechanism capable of incrementally learning unknown classes. This issue was left for future work.

The specific objective #1 (see Chapter1), "To propose a method to describe, in natural language, single and concurrent known events occurring in videos," was achieved through the proposed Target Detection and Localization (TDL) mechanism within the OSVidCap framework. This mechanism detects diverse events that co-occur in a video, allowing the description of all concurrent events detected.

The specific objective #2, "To investigate and devise a method to detect and recognize unseen and unknown events," is an integral aspect of this thesis, as it prevents the model from generating inaccurate descriptions when confronted with unknown events. This goal was accomplished by implementing the "open set module" in the OSVidCap framework, which enables the framework to deal effectively with unknown events. Likewise, the aim of the specific objective #3, "To investigate and devise a method to incrementally learn how to describe the unknown events detected," was also achieved through the approach proposed in Section 3.2. The proposed method enhanced the model's ability to describe new events incrementally.

Moreover, the accomplishment of the specific objective #4, "To create new datasets for training the proposed models, if necessary," was achieved by creating new datasets, as detailed in Sections 4.1.1 and 4.2.1. The lack of adequate datasets in the literature encouraged the creation of these datasets, which have been made publicly available to allow comparisons and benchmarking with other approaches.

Finally, the specific objective #5, "Validate the proposed approaches in public datasets, if applicable," was achieved through a series of experiments conducted with the proposed model, as presented in Chapter 4. These experiments were divided into two parts. The first part, described in Section 4.1, assessed the description of videos containing simple and concurrent human actions within an open-world context. In this context, the model was required to describe human actions occurring in the videos while detecting and disregarding unknown actions to prevent inaccurate descriptions. The obtained results demonstrate the model's ability to describe known actions

accurately and effectively detect and ignore unknown actions, with the model correctly indicating that such action was unknown. The second part of the experiments, described in Section 4.2, aimed to evaluate the proposed video captioning approach in a class incremental learning setting. The results demonstrated the proposed approach's ability to learn new knowledge with minimal or no forgetting of previously learned information. Moreover, the proposed memory expansion method proved essential to the model for acquiring new knowledge, particularly when confronted with numerous new classes.

Despite having achieved the proposed objectives, some areas for improvement were identified during the validation process of the model.

Based on the results and discussions presented, a potential limitation in the proposed approach is the need for more detailed descriptions of the people detected in the videos, such as clothing types or hair color. This challenge arises from the natural variation in word frequency within the training vocabulary, resulting in limited exposure of these specific vocabularies to the model during training. One potential solution to this issue is to employ specialized network models specifically designed to capture such details and to use a post-processing step to incorporate them into the generated descriptions.

Regarding the proposed incremental learning method, a limitation can be cited as the potential demand for memory expansion when acquiring new tasks. These additional computational resources may need to be more sustainable in specific applications. While dynamic memory expansion presents an alternative for such situations, it still requires significant computational resources when learning many new tasks.

Also, the proposed Open-set module in the OSVidCap framework is limited, as it cannot perform incremental learning. This limitation arises from using the EVM, which performs incremental learning by fitting different models to the new data. However, as discussed by Gutoski *et al.* (2021b), the EVM is limited to fixed feature representations. Since we used features from the TI3D model, continuously learning updated (dynamic) feature representations, the EVM cannot be used in an incremental learning context.

Designing a generic method capable of describing a wide range of human actions remains challenging. The complex visual environment and dynamics of temporal structures, including occlusion, fine-grained activities, and tiny-size objects frequently essential for scene description, contribute to the complexity of this video description task.

One of the main barriers in training a generic deep learning model is the need for huge

datasets for the video captioning task. The availability of large-scale datasets with comprehensive annotations is crucial for effectively training such approaches and achieving satisfactory performance levels. Thus, efforts should be focused on creating new datasets and employing semi-automatic annotation methods to mitigate the labor-intensive nature of manual annotation.

Another relevant point is the several metrics commonly used to evaluate video captioning systems. They all have limitations in capturing the nuances of language generation, semantic understanding, and contextual relevance. These metrics often rely on simplistic measures, such as BLEU and METEOR, which assess superficial lexical similarities but need to evaluate the quality and coherence of the generated captions comprehensively. Moreover, these metrics may exhibit high sensitivity to sentence length, leading to biased evaluations in favor of shorter captions. Additionally, existing metrics face challenges in assessing the overall meaning and context of the generated descriptions, making it difficult to distinguish between semantically accurate captions and those merely superficially similar to the ground truth. Thus, developing more sophisticated and context-aware evaluation metrics is essential for a more accurate and comprehensive assessment of video captioning systems.

In conclusion, developing new video description strategies remains essential, especially in an open-world scenario, as these models can adapt and learn from new data over time. Continued research and innovation in this field will pave the way for significant advances in real-world applications, resulting in more effective and accurate video description methods with broader applicability.

5.1 RESEARCH CONTRIBUTIONS AND PUBLISHED PAPERS

Throughout the course of this thesis, many significant contributions have been made and can be summarized as follows:

- A novel video captioning framework to recognize and describe concurrent actions/activities humans perform in an open-set scenario.
- A novel open-set mechanism to detect out-of-domain videos of unseen activities.
- A taxonomy of the existing metrics for video captioning.
- The advantages, shortcomings, and challenges of existing video captioning metrics are identified and discussed.

- A class-incremental learning method for the video captioning task.
- Two novel datasets and protocols for evaluating video captioning in an open-world scenario.

The following paragraphs provide a chronological overview of the published papers that have emerged from this thesis.

In the initial years, I collaborated on works as the main author or co-author to the papers Berno *et al.* (2019), Brilhador *et al.* (2019), Inácio *et al.* (2019), and Inácio *et al.* (2019), which laid the groundwork for exploring ML and DL areas. These early papers provided a foundational understanding and led to the main focus of this thesis – Contributions to Video Captioning in an open-world scenario.

Although unrelated to the thesis's central theme, the paper Inácio e LOPES (2020) enriched my research interests by proposing a method for clothing segmentation. Valuable perspectives were provided that complement the overall understanding of my work and can be explored in future works to enrich the generated description.

Advancing further, papers Inácio *et al.* (2021a) and Inácio *et al.* (2021b) delved into counting people in videos and explaining video anomalies through natural language, respectively. While tangential to the thesis's central theme, they provided insights into video processing and understanding and NLP, both essential topics for this thesis.

The paper Inácio *et al.* (2021) holds a central position in my research as it aligns closely with the thesis's theme. It presents the OSVidCap framework for video captioning in an open-set scenario, presented in Sections 3.1 and 4.1, shaping the direction of my research and guiding subsequent studies.

Furthermore, the effort in Inácio (2023) extended beyond the primary focus of this thesis. This publication presents a comprehensive survey of promising and existing automatic evaluation metrics relevant to the Video Captioning task. Section 2.8.3 presents the examined metrics and discusses the insights acquired from this paper. The paper organizes the existing knowledge in this domain and critically evaluates the advantages and disadvantages of each metric under consideration. This work has significantly enhanced the comprehension of evaluation metrics, allowing for an expansion of the research scope. Furthermore, it has laid the foundation for future works on this topic.

Furthermore, a paper focusing on Incremental Learning Video Captioning is currently in progress. This work aims to present the method described in Section 3.2 and the results presented in Section 4.2, holding great potential to significantly contribute to the existing body

of knowledge in the field. Rigorous data analysis and experimentation are employed and provide valuable insights into Incremental Learning in Video Captioning

5.2 FUTURE WORK

During the development of this thesis, several areas for potential improvement and further investigation have emerged. Future research should focus on expanding the proposed approach to enhance the quality and comprehensiveness of the generated descriptions. This enrichment is crucial in applications requiring detailed information regarding humans and objects, such as surveillance and human monitoring. The primary challenge in this effort lies in extracting visual concepts essential for enhancing the description. For instance, soft biometric traits can be incorporated to boost human descriptions, while attributes such as color, make, and model can be used to enhance vehicle descriptions. To address this challenge, off-the-shelf models could be employed to extract specific information. Thus, a module within the OSVidCap framework could be developed to integrate this detailed information into the overall sentence structure, ensuring high fluency and grammatical accuracy.

Future research should also focus on improving the open-set module in OSVidCap to perform incremental learning. Also, recent approaches have been proposed to solve the EVM limitations in this context (GUTOSKI *et al.*, 2021b; KOCH *et al.*, 2022) and can be further explored to enhance the adaptability and performance of the system.

Future research should also focus on developing new datasets for video captioning in an open-world environment. The lack of high-quality datasets for this specific scenario might restrict the exploration and advancement of new methods and applications. Additionally, most existing datasets contain only English captions, while datasets with captions described in other languages are very scarce. Thus, future work could also consider designing a video captioning dataset in Portuguese.

The evaluation of video description systems commonly relies on metrics based on n-gram overlap to measure the similarity between the generated and reference sentences. However, these metrics face difficulties in assessing the sentences' semantic aspects. EMScore is the only metric specifically proposed for the video description task, aiming to measure the similarity between a sentence and the visual content of a video. However, it has been found to have certain drawbacks, as outlined by Inácio (2023). Therefore, further research should develop novel metrics that focus on the semantic analysis of the generated sentences.

In recent years, the Transformer architecture has been successfully explored, demonstrating superior performance across various natural language processing tasks. As a potential avenue for future investigation, researchers may explore using transformers for the video captioning task in a class-incremental learning fashion. This line of investigation could also involve exploring the use of transformer architectures to combine visual and audio information from videos through multimodal fusion techniques. Additionally, researchers can explore class-incremental learning (CIL) approaches, including dynamic architectures, rehearsal methods, or knowledge distillation, to adapt transformer models for class-incremental learning. By conducting further research in this area, researchers can investigate and develop effective strategies for integrating transformer models into video captioning tasks in a class-incremental learning scenario, thereby enhancing the accuracy and quality of the generated captions.

BIBLIOGRAPHY

AAFAQ, N.; AKHTAR, N.; LIU, W. *et al.* Spatio-temporal dynamics and semantic attribute enriched visual encoding for video captioning. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Snowmass, Colorado: *[s.n.]*, 2019. p. 12487–12496.

AAFAQ, N.; AKHTAR, N.; LIU, W.; MIAN, A. Empirical autopsy of deep video captioning encoder-decoder architecture. **Array**, Elsevier, v. 9, p. 100052, 2021.

AAFAQ, N.; MIAN, A.; LIU, W.; GILANI, S. Z.; SHAH, M. Video description: A survey of methods, datasets, and evaluation metrics. **ACM Computing Surveys**, v. 52, n. 6, p. 37, 2019.

AHMED, S.; SAIF, A. F. M.; HANIF, M. I. *et al.* Att-BiL-SL: Attention-based Bi-LSTM and sequential LSTM for describing video in the textual formation. **Applied Sciences**, v. 12, n. 1, p. 317, Dec 2021.

AMARESH, M.; CHITRAKALA, S. Video captioning using deep learning: An overview of methods, datasets and metrics. *In*: **Proc. of the International Conference on Communication and Signal Processing**. Chennai, India: *[s.n.]*, 2019. p. 656–661.

AMIRIAN, S.; RASHEED, K.; TAHA, T. R.; ARABNIA, H. R. Automatic image and video caption generation with deep learning: A concise review and algorithmic overlap. **IEEE Access**, v. 8, p. 218386–218400, 2020.

ANDERSON, P.; FERNANDO, B.; JOHNSON, M.; GOULD, S. SPICE: Semantic propositional image caption evaluation. *In*: **Proc. of the European Conference on Computer Vision (ECCV)**. Cham: Springer International Publishing, 2016. p. 382–398.

BABARIYA, R. J.; TAMAKI, T. Meaning guided video captioning. *In*: **Proc. of the Asian Conference on Pattern Recognition**. Auckland, New Zealand: *[s.n.]*, 2019. p. 478–488.

BAI, Y.; WANG, J.; LONG, Y. *et al.* Discriminative latent semantic graph for video captioning. *In*: **Proc. of the 29th ACM International Conference on Multimedia**. New York, NY, USA: *[s.n.]*, 2021. p. 3556–3564.

BAKAROV, Amir. A survey of word embeddings evaluation methods. **preprint arXiv:1801.09536**, 2018.

BARALDI, L.; GRANA, C.; CUCCHIARA, R. Hierarchical boundary-aware neural encoder for video captioning. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Honolulu, Hawaii: *[s.n.]*, 2017. p. 1657–1666.

BARATI, E.; CHEN, X. Critic-based attention network for event-based video captioning. *In*: **Proc. of the 27th ACM International Conference on Multimedia**. New York, NY, USA: ACM, 2019. p. 811–817.

BARBU, A.; BRIDGE, A.; BURCHILL, Z. *et al.* Video in sentences out. *In*: **Proc. of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence**. Arlington, Virginia, USA: *[s.n.]*, 2012. p. 102–112.

BELOUADAH, E.; POPESCU, A.; KANELLOS, I. A comprehensive study of class incremental learning algorithms for visual tasks. **Neural Networks**, Elsevier, v. 135, p. 38–54, 2021.

BENDALE, Abhijit; BOULT, Terrance. Towards open world recognition. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. *[S.l.: s.n.]*, 2015.

BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. **IEEE Transactions on Neural Networks**, v. 5, n. 2, p. 157–166, 1994.

BERNO, B. C. S.; GABARDO, A. C.; HATTORI, L. T.; GUTOSKI, M.; INÁCIO, A. S.; LOPES, H. S. A framework for analyzing book covers and co-purchases using object detection and data mining methods. *In*: **Proc. of the IEEE Latin American Conference on Computational Intelligence (LA-CCI)**. *[S.l.: s.n.]*, 2019. p. 1–6.

BIN, Y.; YANG, Y.; SHEN, F.; XIE, N.; SHEN, H. T.; LI, X. Describing video with attention-based bidirectional LSTM. **IEEE Transactions on Cybernetics**, v. 49, n. 7, p. 2631–2641, 2019.

BOCHKOVSKIY, A.; WANG, C.; LIAO, H. M. YOLOv4: Optimal speed and accuracy of object detection. **preprint arXiv:2004.10934**, p. 1–17, 2020.

BOJANOWSKI, P.; GRAVE, E.; JOULIN, A.; MIKOLOV, T. Enriching word vectors with subword information. **Transactions of the Association for Computational Linguistics**, v. 5, p. 135–146, 2017.

BRILHADOR, A.; GUTOSKI, M.; HATTORI, L. T.; INÁCIO, A. S.; LAZZARETTI, A. E.; LOPES, H. S. Classification of weeds and crops at the pixel-level using convolutional neural networks and data augmentation. *In*: **Proc. of the IEEE Latin American Conference on Computational Intelligence (LA-CCI)**. *[S.l.: s.n.]*, 2019. p. 1–6.

BRUNELLI, R. **Template Matching Techniques in Computer Vision: theory and practice**. Chichester, U.K: John Wiley & Sons, 2009.

BUCH, S.; ESCORCIA, V.; SHEN, C.; GHANEM, B.; NIEBLES, J. C. SST: Single-Stream Temporal Action Proposals. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Honolulu, Hawaii: *[s.n.]*, 2017. p. 6373–6382.

BUDVYTIS, I.; TEICHMANN, M.; VOJIR, T.; CIPOLLA, R. Large scale joint semantic re-localisation and scene understanding via globally unique instance coordinate regression. *In*: **Proc. of the 30th British Machine Vision Conference**. *[S.l.: s.n.]*, 2019. p. 1–13.

CAO, Z.; SIMON, T.; WEI, S.; SHEIKH, Y. Realtime multi-person 2D pose estimation using part affinity fields. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. *[S.l.: s.n.]*, 2017. p. 7291–7299.

CARREIRA, J.; ZISSERMAN, A. Quo vadis, action recognition? a new model and the kinetics dataset. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Honolulu, USA: *[s.n.]*, 2017. p. 4724–4733.

CARRIO, A.; SAMPEDRO, C.; RODRIGUEZ-RAMOS, A.; CAMPOY, P. A review of deep learning methods and applications for unmanned aerial vehicles. **Journal of Sensors**, v. 2017, p. 1–13, 2017.

CHAUDHRY, A.; DOKANIA, P. K.; AJANTHAN, T.; TORR, P. H. S. Riemannian walk for incremental learning: Understanding forgetting and intransigence. *In*: **Proc. of the European Conference on Computer Vision (ECCV)**. *[S.l.: s.n.]*, 2018. p. 532–547.

CHEN, D.; DOLAN, W. Collecting highly parallel data for paraphrase evaluation. *In*: **Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies**. Portland, Oregon: *[s.n.]*, 2011. p. 190–200.

CHEN, H.; LI, J.; HU, X. Delving deeper into the decoder for video captioning. *In*: **Proc. of the 24th European Conference on Artificial Intelligence (ECAI)**. *[S.l.: s.n.]*, 2020. p. 1–8.

CHEN, H.; LIN, K.; MAYE, A. *et al.* A semantics-assisted video captioning model trained with scheduled sampling. **Frontiers in Robotics and AI**, v. 7, p. 475767, 2020.

CHEN, M.; LI, Y.; ZHANG, Z.; HUANG, S. TVT: Two-view transformer network for video captioning. *In*: **Proc. of The 10th Asian Conference on Machine Learning**. Beijing, China: *[s.n.]*, 2018. p. 847–862.

CHEN, S.; JIANG, Y. Motion guided spatial attention for video captioning. *In*: **Proc. of the 33rd AAAI Conference on Artificial Intelligence**. Honolulu, Hawaii: *[s.n.]*, 2019. v. 33, p. 8191–8198.

CHEN, S.; JIN, Q.; CHEN, J.; HAUPTMANN, A. G. Generating video descriptions with latent topic guidance. **IEEE Transactions on Multimedia**, v. 21, n. 9, p. 2407–2418, 2019.

CHEN, S.; ZHONG, X.; WU, S. *et al.* Memory-attended semantic context-aware network for video captioning. **Soft Computing**, p. 1–13, 2021.

CHEN, T.; ZHAO, Q.; SONG, J. Boundary detector encoder and decoder with soft attention for video captioning. *In*: J., Song; X., Zhu (Ed.). **Web and Big Data**. Cham: Springer International Publishing, 2019. p. 105–115.

CHEN, X.; FANG, H.; LIN, T. *et al.* Microsoft COCO captions: Data collection and evaluation server. **preprint arXiv:1504.00325**, p. 1–7, 2015.

CHEN, X.; SONG, J.; ZENG, P. *et al.* Support-set based multi-modal representation enhancement for video captioning. *In*: **Proc. IEEE International Conference on Multimedia and Expo (ICME)**. *[S.l.: s.n.]*, 2022. p. 1–6.

CHEN, Y-C.; LI, L.; YU, L. *et al.* UNITER: UNiversal Image-TExt Representation Learning. *In*: SPRINGER. **Proc. of the European Conference on Computer Vision (ECCV)**. *[S.l.]*, 2020. p. 104–120.

CHEN, Y.; WANG, S.; ZHANG, W. *et al.* Less Is More: Picking informative frames for video captioning. *In*: **Proc. of the European Conference on Computer Vision (ECCV)**. Munich, Germany: *[s.n.]*, 2018. p. 367–384.

CHEN, Y.; ZHANG, W.; WANG, S. *et al.* Saliency-based spatiotemporal attention for video captioning. *In*: **Proc. of the IEEE Fourth International Conference on Multimedia Big Data**. Xi'an, China: *[s.n.]*, 2018. p. 1–8.

CHEN, Z.; LIU, B. **Lifelong Machine Learning**. 2. ed. San Rafael, CA: Morgan & Claypool Publishers, 2018.

CHERIAN, A.; WANG, J.; HORI, C.; MARKS, T. Spatio-temporal ranked-attention networks for video captioning. *In*: **Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)**. Snowmass, Colorado: *[s.n.]*, 2020. p. 1617–1626.

CUI, Y.; YANG, G.; VEIT, A. *et al.* Learning to evaluate image captioning. *In*: **Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. *[S.l.: s.n.]*, 2018. p. 5804–5812.

DASKALAKIS, E.; TZELEPI, M.; TEFAS, A. Learning deep spatiotemporal features for video captioning. **Pattern Recognition Letters**, v. 116, p. 143 – 149, 2018.

De LANGE, M.; ALJUNDI, R.; MASANA, M. *et al.* A continual learning survey: Defying forgetting in classification tasks. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 44, n. 7, p. 3366–3385, 2022.

DEEPU, S.; RAJ, P.; RAJARAAJESWARI, S. A framework for text analytics using the bag of words (BoW) model for prediction. *In*: **Proc. of the 1st International Conference on Innovations in Computing & Networking (ICICN)**. *[S.l.: s.n.]*, 2016. p. 12–13.

DEL CHIARO, R.; TWARDOWSKI, B.; BAGDANOV, A.; Van De Weijer, J. RATT: Recurrent attention to transient tasks for continual image captioning. *In*: **Proc. of the 34th Conference on Neural Information Processing Systems (NeurIPS)**. *[S.l.: s.n.]*, 2020.

DENG, C.; CHEN, S.; CHEN, D.; HE, Y.; WU, Q. Sketch, Ground, and Refine: Top-Down Dense Video Captioning. *In*: **Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. *[S.l.: s.n.]*, 2021. p. 234–243.

DENG, J.; LI, L.; ZHANG, B. *et al.* Syntax-guided hierarchical attention network for video captioning. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 32, n. 2, p. 880–892, 2022.

DENKOWSKI, M.; LAVIE, A. Choosing the right evaluation for machine translation: an examination of annotator and automatic metric performance on human judgment tasks. *In*: **Proc. of the Ninth Conference of the Association for Machine Translation in the Americas**. Denver, Colorado: *[s.n.]*, 2010. p. 1–9.

DEVLIN, J.; CHANG, M-W.; LEE, K.; TOUTANOVA, K. BERT: pre-training of deep bidirectional transformers for language understanding. *In*: **Proc. of the North American Chapter of the Association for Computational Linguistics**. *[S.l.: s.n.]*, 2019. p. 4171–4186.

DIETTERICH, T. Overfitting and undercomputing in machine learning. **ACM Computing Surveys**, v. 27, n. 3, p. 326–327, 1995.

DILAWARI, A.; KHAN, M. U. G. ASoVS: Abstractive summarization of video sequences. **IEEE Access**, v. 7, n. 1, p. 29253–29263, 2019.

DIWAN, T.; ANIRUDH, G.; TEMBHURNE, J. V. Object detection using YOLO: challenges, architectural successors, datasets and applications. **Multimedia Tools and Applications**, v. 82, p. 9243–9275, 2022.

DONG, J.; GAO, K.; CHEN, X.; CAO, J. Refocused attention: Long short-term rewards guided video captioning. **Neural Processing Letters**, v. 52, n. 2, p. 1–14, 2019.

DU, X.; YUAN, J.; HU, L.; DAI, Y. Description generation of open-domain videos incorporating multimodal features and bidirectional encoder. **The Visual Computer**, v. 35, n. 12, p. 1703–1712, 2019.

DUAN, X.; HUANG, W.; GAN, C. *et al.* Weakly supervised dense event captioning in videos. *In*: **Proc. of the 31th Conference on Neural Information Processing Systems (NeurIPS)**. Montreal, Canadá: *[s.n.]*, 2018. p. 3059–3069.

ESCORCIA, V.; HEILBRON, F. C.; NIEBLES, J. C.; GHANEM, B. DAPs: Deep action proposals for action understanding. *In*: **Proc. of the European Conference on Computer Vision (ECCV)**. Amsterdam: *[s.n.]*, 2016. p. 768–784.

FANG, K.; ZHOU, L.; JIN, C.; ZHANG, Y.; WENG, K.; ZHANG, T.; FAN, W. Fully convolutional video captioning with coarse-to-fine and inherited attention. *In*: **Proc. of the 33rd AAAI Conference on Artificial Intelligence**. Honolulu, Hawaii: *[s.n.]*, 2019. p. 8271–8278.

FANG, Z.; GOKHALE, T.; BANERJEE, P.; BARAL, C.; YANG, Y. Video2Commonsense: Generating commonsense descriptions to enrich video captioning. *In*: **Proc. of Conference on Empirical Methods in Natural Language Processing (EMNLP)**. *[S.l.]*: Association for Computational Linguistics, 2020. p. 840–860.

FEINGLASS, J.; YANG, Y. SMURF: SeMantic and linguistic UndeRstanding fusion for caption evaluation via typicality analysis. *In*: **Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**. *[S.l.: s.n.]*, 2021. p. 2250–2260.

FRANCIS, D.; HUET, B. L-STAP: Learned spatio-temporal adaptive pooling for video captioning. *In*: **Proc. of the 1st ACM International Workshop on AI for Smart TV Content Production, Access and Delivery**. Nice, France: *[s.n.]*, 2019. p. 33–41.

FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. **Biological Cybernetics**, v. 36, p. 193–202, 1980.

GAO, L.; GUO, Z.; ZHANG, H.; XU, X.; SHEN, H. T. Video captioning with attention-based LSTM and semantic consistency. **IEEE Transactions on Multimedia**, v. 19, n. 9, p. 2045–2055, 2017.

GAO, L.; LI, X.; SONG, J. *et al.* Hierarchical LSTMs with adaptive attention for visual captioning. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 42, n. 5, p. 1112–1131, 2020.

GAO, L.; WANG, X.; SONG, J.; LIU, Y. Fused GRU with semantic-temporal attention for video captioning. **Neurocomputing**, v. 395, p. 222–228, 2020.

GELLA, S.; LEWIS, M.; ROHRBACH, M. A dataset for telling the stories of social media videos. *In*: **Proc. of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. *[S.l.: s.n.]*, 2018. p. 968–974.

GENG, C.; HUANG, S-J.; CHEN, S. Recent advances in open set recognition: A survey. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 43, n. 10, p. 3614–3631, 2021.

GENG, C.; TAO, L.; CHEN, S. Guided CNN for generalized zero-shot and open-set recognition using visual and semantic prototypes. **Pattern Recognition**, v. 102, p. 107263, 2020.

GENG, X.; SMITH-MILES, K. Incremental learning. *In*: LI, S. Z.; JAIN, A. (Ed.). **Encyclopedia of Biometrics**. Boston, MA: Springer US, 2009. p. 731–735.

GEPPERTH, A.; HAMMER, B. Incremental learning algorithms and applications. *In*: **European Symposium on Artificial Neural Networks (ESANN)**. Bruges, Belgium: *[s.n.]*, 2016. p. 1–12.

GERS, F. A.; SCHMIDHUBER, J.; CUMMINS, F. Learning to forget: continual prediction with LSTM. *In*: **Proc. of the Ninth International Conference on Artificial Neural Networks**. Edinburgh, UK: *[s.n.]*, 1999. p. 850–855.

GHADERI, Z.; SALEWSKI, L.; LENSCH, H. P. A. Diverse video captioning by adaptive spatio-temporal attention. *In*: BJÖRN, A.; BERNARD, F. *et al.* (Ed.). **Pattern Recognition**. Cham: *[s.n.]*, 2022. p. 409–425.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge, MA, USA: MIT Press, 2017.

GUADARRAMA, S.; KRISHNAMOORTHY, N.; MALKARNENKAR, G. *et al.* YouTube2Text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. *In*: **Proc. of the IEEE International Conference on Computer Vision (ICCV)**. Sydney, Australia: *[s.n.]*, 2013. p. 2712–2719.

GUO, Y.; ZHANG, J.; GAO, L. Exploiting long-term temporal dynamics for video captioning. **World Wide Web**, v. 22, n. 2, p. 735–749, 2019.

GUTOSKI, M.; LAZZARETTI, A. E.; LOPES, H. S. Deep metric learning for open-set human action recognition in videos. **Neural Computing and Applications**, Springer, v. 33, p. 1207–1220, 2021.

GUTOSKI, Matheus; LAZZARETTI, André Eugenio; LOPES, Heitor Silvério. Incremental human action recognition with dual memory. **Image and Vision Computing**, Elsevier, v. 116, p. 104313, 2021.

HARA, K.; KATAOKA, H.; SATOH, Y. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. *[S.l.: s.n.]*, 2018.

HAYKIN, S. **Neural networks: A Comprehensive Foundation**. 2. ed. Delhi, India: Pearson Prentice Hall, 1999.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Las Vegas, Nevada: *[s.n.]*, 2016. p. 770–778.

HEILBRON, F. C.; ESCORCIA, V.; GHANEM, B.; NIEBLES, J. C. ActivityNet: A large-scale video benchmark for human activity understanding. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Boston, USA: *[s.n.]*, 2015. p. 961–970.

HEMALATHA, M.; SEKHAR, C. C. Domain-specific semantics guided approach to video captioning. *In*: **Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)**. Snowmass, Colorado: *[s.n.]*, 2020. p. 1576–1585.

HESSEL, J.; HOLTZMAN, A.; FORBES, M. *et al.* CLIPScore: A reference-free evaluation metric for image captioning. **preprint arXiv:2104.08718**, 2021.

HG, S.; S, M. Enhanced data processing system for real time video monitoring. *In*: **Proc. of the International Conference on Electronics and Sustainable Communication Systems (ICESC)**. Coimbatore, India: *[s.n.]*, 2020. p. 526–533.

HIRSCHBERG, J.; MANNING, C. D. Advances in natural language processing. **Science**, v. 349, n. 6245, p. 261–266, 2015.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, 1997.

HOU, J.; WU, X.; ZHAO, W. *et al.* Joint syntax representation learning and visual cue translation for video captioning. *In*: **Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)**. Seoul: *[s.n.]*, 2019. p. 8917–8926.

HU, Y.; CHEN, Z.; ZHA, Z.; WU, F. Hierarchical global-local temporal modeling for video captioning. *In*: **Proc. of the 27th ACM International Conference on Multimedia**. New York, NY, USA: *[s.n.]*, 2019. p. 774–783.

IASHIN, V.; RAHTU, E. Multi-modal dense video captioning. *In*: **Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)**. *[S.l.: s.n.]*, 2020. p. 4117–4126.

INÁCIO, A. S.; BRILHADOR, A.; LOPES, H. S. Semantic segmentation of clothes in the context of soft biometrics using deep learning methods. *In*: **Anais do 14^o Congresso Brasileiro de Inteligência Computacional**. Curitiba, PR: *[s.n.]*, 2019. p. 1–7.

INÁCIO, A. S.; GUTOSKI, M.; LAZZARETTI, A. E.; LOPES, H. S. OSVidCap: A framework for the simultaneous recognition and description of concurrent actions in videos in an open-set scenario. **IEEE Access**, v. 9, p. 137029–137041, 2021.

INÁCIO, A. S.; HATTORI, L.; GUTOSKI, M.; L., A.; LOPES, H. S. Análise da fragmentação partidária na Assembleia Legislativa do Rio Grande do Sul com métodos de mineração de dados. *In*: **Proc. of the VIII Brazilian Workshop on Social Network Analysis and Mining**. Belém, PA: *[s.n.]*, 2019. p. 161–166.

INÁCIO, A. S.; LOPES, H. S. EPYNET: Efficient pyramidal network for clothing segmentation. **IEEE Access**, v. 8, p. 187882–187892, 2020.

INÁCIO, A. S.; RAMOS, R. H.; LOPES, H. S. Deep learning for people counting in videos by age and gender. *In*: **Anais do 15^o Congresso Brasileiro de Inteligência Computacional**. Joinville, SC: *[s.n.]*, 2021. p. 1–6.

INÁCIO, A. S.; TEIXEIRA, R. M.; LOPES, H. S. Explainable anomaly detection in videos based on the description of atomic actions. *In*: **Anais do 15^o Congresso Brasileiro de Inteligência Computacional**. Joinville, SC: *[s.n.]*, 2021. p. 1–6.

INÁCIO, H. S. Lopes A. S. Evaluation metrics for video captioning: a survey. **Machine Learning With Applications**, v. 13, n. 100488, p. 1–17, 2023.

ISLAM, S.; DASH, A.; SEUM, A. *et al.* Exploring video captioning techniques: A comprehensive survey on deep learning methods. **SN Computer Science**, v. 2, n. 2, p. 1–28, 2021.

JAHNE, B.; HAUßECKER, H. **Computer Vision and Applications**: A guide for students and practitioners. USA: Academic Press., 2000.

JAIN, N. K.; SAINI, R. K.; MITTAL, P. A review on traffic monitoring system techniques. *In*: RAY, K.; SHARMA, T. K.; RAWAT, S. *et al.* (Ed.). **Soft Computing: Theories and Applications**. Singapore: Springer, 2019. p. 569–577.

JELODAR, A. B.; PAULIUS, D.; SUN, Y. Long activity video understanding using functional object-oriented network. **IEEE Transactions on Multimedia**, v. 21, n. 7, p. 1813–1824, 2019.

JI, L.; TU, R.; LIN, K. *et al.* Multimodal graph neural network for video procedural captioning. **Neurocomputing**, v. 488, p. 88–96, 2022.

JI, W.; WANG, R. A multi-instance multi-label dual learning approach for video captioning. **ACM Transactions on Multimedia Computing, Communications, and Applications**, v. 17, n. 2s, jun 2021.

JI, W.; WANG, R.; TIAN, Y.; WANG, X. An attention-based dual learning approach for video captioning. **Applied Soft Computing**, v. 117, n. 108332, p. 9, 2022.

JIANG, M.; HU, J.; HUANG, Q. *et al.* REO-Relevance, Extraness, Omission: A fine-grained evaluation for image captioning. *In*: **Proc. of 10th International Joint Conference on Natural Language Processing**. *[S.l.: s.n.]*, 2019. p. 1475–1480.

JIANG, M.; HUANG, Q.; ZHANG, L. *et al.* Tiger: Text-to-image grounding for image caption evaluation. *In*: **Proc. 9th International Joint Conference on Natural Language Processing**. *[S.l.: s.n.]*, 2020. p. 2141–2152.

JIN, Q.; CHEN, J.; CHEN, S. *et al.* Describing videos using multi-modal fusion. *In*: **Proc. of the 24th ACM International Conference on Multimedia**. New York, NY, USA: *[s.n.]*, 2016. p. 1087–1091.

JIN, T.; HUANG, S.; CHEN, M. *et al.* SBAT: Video captioning with sparse boundary-aware transformer. *In*: **Proc. of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)**. *[S.l.: s.n.]*, 2021. p. 1–7.

JIN, T.; LI, Y.; ZHANG, Z. Recurrent convolutional video captioning with global and local attention. **Neurocomputing**, v. 370, p. 118 – 127, 2019.

JIN, T.; ZHAO, Z.; WANG, P. *et al.* Interaction augmented transformer with decoupled decoding for video captioning. **Neurocomputing**, v. 492, p. 496–507, 2022.

KARPATHY, A.; TODERICI, G.; SHETTY, S. *et al.* Large-scale video classification with convolutional neural networks. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Columbus, Ohio: *[s.n.]*, 2014. p. 1725–1732.

KAY, W.; CARREIRA, J.; SIMONYAN, K. *et al.* The kinetics human action video dataset. **preprint arXiv:1705.06950**, 2017.

KILICKAYA, M.; ERDEM, A.; IKIZLER-CINBIS, N.; ERDEM, E. Re-evaluating automatic metrics for image captioning. *In*: **Proc. of the 15th Conference of the European Chapter of the Association for Computational Linguistics**. *[S.l.: s.n.]*, 2017. v. 1, p. 199–209.

KIRKPATRICK, J.; PASCANU, R.; RABINOWITZ, N. *et al.* Overcoming catastrophic forgetting in neural networks. **Proc. of the National Academy of Sciences**, v. 114, n. 13, p. 3521–3526, 2017.

KITCHENHAM, B. **Procedures for Performing Systematic Reviews**. Australia, 2004. 1–26 p. Technical Report TR/SE-0401.

KOCH, T.; LIEBEZEIT, F.; RIESS, C. *et al.* Exploring the open world using incremental extreme value machines. *In*: **Proc. of the 26th International Conference on Pattern Recognition (ICPR)**. *[S.l.: s.n.]*, 2022. p. 2792–2799.

KONG, Y.; FU, Y. Human action recognition and prediction: A survey. **International Journal of Computer Vision**, v. 130, n. 5, p. 1366–1401, 2022.

KRISHNA, R.; HATA, K.; REN, F. *et al.* Dense-captioning events in videos. *In*: **Proc. of the IEEE International Conference on Computer Vision (ICCV)**. Venice, Italy: *[s.n.]*, 2017. p. 706–715.

KRISHNAMOORTHY, N.; MALKARNENKAR, G.; MOONEY, R. *et al.* Generating natural-language video descriptions using text-mined knowledge. **Proc. of the Twenty-Seventh AAAI Conference on Artificial Intelligence**, v. 27, n. 1, p. 541–547, Jun. 2013.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet classification with deep convolutional neural networks. *In*: **Proc. of the 25th Conference on Neural Information Processing Systems (NeurIPS)**. Lake Tahoe, NV: Curran Associates Inc., 2012. v. 1, p. 1097–1105.

L., Yinhan; O., Myle; G., Naman *et al.* RoBERTa: a robustly optimized BERT pretraining approach. **ArXiv**, abs/1907.11692, 2019.

LAINA, I.; RUPPRECHT, C.; NAVAB, N. Towards unsupervised image captioning with shared multimodal embeddings. *In*: **Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)**. *[S.l.: s.n.]*, 2019.

LAVEE, G.; RIVLIN, E.; RUDZSKY, M. Understanding video events: a survey of methods for automatic interpretation of semantic occurrences in video. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, IEEE, v. 39, n. 5, p. 489–504, 2009.

LAVIE, A.; AGARWAL, A. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. *In*: **Proc. of the Second Workshop on Statistical Machine Translation**. Prague, Czech Republic: Association for Computational Linguistics, 2007. p. 228–231.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998.

Lee, H.; Kim, I. Generating natural video descriptions using semantic gate. *In*: **Proc. of the International Joint Conference on Neural Networks (IJCNN)**. *[S.l.: s.n.]*, 2019. p. 1–7.

LEE, H.; YOON, S.; DERNONCOURT, F. *et al.* ViLBERTScore: Evaluating image caption using vision-and-language BERT. *In*: **Proc. of the 1st Workshop on Evaluation and Comparison of NLP Systems**. *[S.l.: s.n.]*, 2020. p. 34–39.

LEE, H.; YOON, S.; DERNONCOURT, F. *et al.* UMIC: An unreferenced metric for image captioning via contrastive learning. *In*: **Proc. of the 11th International Joint Conference on Natural Language Processing**. *[S.l.: s.n.]*, 2021. v. 2, p. 220–226.

LEE, J. Y. Deep multimodal embedding for video captioning. **Multimedia Tools and Applications**, v. 78, n. 22, p. 31793–31805, 2019.

LEI, J.; YU, L.; BERG, T. L.; BANSAL, M. TVR: A large-scale dataset for video-subtitle moment retrieval. *In*: SPRINGER. **Proc. of the European Conference on Computer Vision (ECCV)**. *[S.l.]*, 2020. p. 447–463.

LI, H.; SONG, D.; LIAO, L.; PENG, C. REVnet: Bring reviewing into video captioning for a better description. *In*: **Proc. of the IEEE International Conference on Multimedia and Expo (ICME)**. Shanghai, China: *[s.n.]*, 2019. p. 1312–1317.

LI, J.; XIE, X.; PAN, Q. *et al.* SGM-Net: Skeleton-guided multimodal network for action recognition. **Pattern Recognition**, v. 104, p. 107356, 2020.

LI, L.; GAO, X.; DENG, J. *et al.* Long short-term relation transformer with global gating for video captioning. **IEEE Transactions on Image Processing**, v. 31, p. 2726–2738, 2022.

LI, L.; GONG, B. End-to-end video captioning with multitask reinforcement learning. *In*: **Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)**. Waikoloa Village, Hawaii: *[s.n.]*, 2019. p. 339–348.

LI, L.; ZHANG, Y.; TANG, S. *et al.* Adaptive spatial location with balanced loss for video captioning. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 32, n. 1, p. 17–30, 2022.

LI, Ping; ZHANG, Pan; XU, Xianghua. Graph convolutional network meta-learning with multi-granularity pos guidance for video captioning. **Neurocomputing**, v. 472, p. 294–305, 2022.

LI, X.; ZHOU, Z.; CHEN, L.; GAO, L. Residual attention-based LSTM for video captioning. **World Wide Web**, Springer, v. 22, n. 2, p. 621–636, 2019.

LI, Yang; YANG, Tao. Word embedding for understanding natural language: a survey. *In*: SRINIVASAN, S. (Ed.). **Guide to big data applications**. *[S.l.]*: Springer, 2018. p. 83–104.

LI, Y.; YAO, T.; PAN, Y.; CHAO, H.; MEI, T. Jointly localizing and describing events for dense video captioning. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Salt Lake City, Utah: *[s.n.]*, 2018. p. 7492–7500.

LI, Z.; HOIEM, D. Learning without forgetting. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 40, n. 12, p. 2935–2947, 2018.

LIN, C. ROUGE: A package for automatic evaluation of summaries. *In*: **Proc. of the Workshop on Text Summarization Branches Out**. Barcelona, Spain: *[s.n.]*, 2004. p. 74–81.

LIN, K.; LI, L.; LIN, C. C. *et al.* SwinBERT: End-to-end transformers with sparse attention for video captioning. *In*: **Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. *[S.l.: s.n.]*, 2022. p. 17949–17958.

LIPTON, Z. C.; BERKOWITZ, J.; ELKAN, C. A critical review of recurrent neural networks for sequence learning. **preprint arXiv: 1506.00019**, p. 1–38, 2015.

LIU, A.; QIU, Y.; WONG, Y.; SU, Y.; KANKANHALLI, M. A fine-grained spatial-temporal attention model for video captioning. **IEEE Access**, v. 6, p. 68463–68471, 2018.

LIU, A.; XU, N.; WONG, Y.; LI, J.; SU, Y.; KANKANHALLI, M. Hierarchical & multimodal video captioning: Discovering and transferring multimodal knowledge for vision to language. **Computer Vision and Image Understanding**, v. 163, p. 113 – 125, 2017.

LIU, L.; OUYANG, W.; WANG, X.; FIEGUTH, P.; CHEN, J.; LIU, X.; PIETIKÄINEN, M. Deep learning for generic object detection: A survey. **International Journal of Computer Vision**, Springer, v. 128, n. 2, p. 261–318, 2020.

LIU, Sheng; REN, Zhou; YUAN, Junsong. SibNet: Sibling convolutional encoder for video captioning. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 43, n. 9, p. 3259–3272, 2021.

LOPEZ-PAZ, D.; RANZATO, M. A. Gradient episodic memory for continual learning. *In*: GUYON, I.; U.von Luxburg; BENGIO, S. *et al.* (Ed.). **Proc. of the 31th Conference on Neural Information Processing Systems ((NeurIPS)**. *[S.l.]*: Curran Associates, Inc., 2017. v. 30.

LU, J.; BATRA, D.; PARIKH, D.; LEE, S. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *In*: **Proc. of the Advances in Seural Information Processing Systems (NeurIPS)**. *[S.l.: s.n.]*, 2019. v. 32.

LU, J.; LIU, A.; DONG, F. *et al.* Learning under concept drift: A review. **IEEE Transactions on Knowledge and Data Engineering**, v. 31, n. 12, p. 2346–2363, 2019.

LU, Xuyang; GAO, Yang. Guide and interact: scene-graph based generation and control of video captions. **Multimedia Systems**, Springer, p. 1–13, 2022.

MADHYASTHA, P. S.; WANG, J.; SPECIA, L. VIFIDEL: Evaluating the visual fidelity of image descriptions. *In*: **Proc. of the 57th Annual Meeting of the Association for Computational Linguistics**. *[S.l.: s.n.]*, 2019. p. 6539–6550.

MAI, Z.; LI, R.; JEONG, J. *et al.* Online continual learning in image classification: An empirical survey. **Neurocomputing**, Elsevier, v. 469, p. 28–51, 2022.

MAN, X.; OUYANG, D.; LI, X. *et al.* Scenario-aware recurrent transformer for goal-directed video captioning. **ACM Transactions on Multimedia Computing, Communications, and Applications**, v. 18, n. 4, p. 1–17, 2022.

MARUF, S.; SALEH, F.; HAFFARI, G. A survey on document-level neural machine translation: Methods and evaluation. **ACM Computing Surveys**, v. 54, n. 2, p. 1–36, mar 2021.

MASANA, M.; LIU, X.; TWARDOWSKI, B.; MENTA, M.; BAGDANOV, A. D.; van de WEIJER, J. Class-incremental learning: Survey and performance evaluation on image classification. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 45, n. 5, p. 5513–5533, 2023.

McCLOSKEY, M.; COHEN, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. **Psychology of Learning and Motivation**, Elsevier, v. 24, p. 109–165, 1989.

MERMILLOD, M.; BUGAISKA, A.; BONIN, P. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. **Frontiers in Psychology**, v. 4, p. 504, 2013.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. **preprint arXiv:1301.3781**, 2013.

MIKOLOV, T; GRAVE, E.; BOJANOWSKI, P.; PUHRSCH, C.; JOULIN, A. Advances in pre-training distributed word representations. *In*: **Proc. of the Eleventh International Conference on Language Resources and Evaluation (LREC)**. Miyazaki, Japan: *[s.n.]*, 2018. p. 52–55.

MUN, J.; YANG, L.; REN, Z.; XU, N.; HAN, B. Streamlined dense video captioning. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Long Beach, CA: *[s.n.]*, 2019. p. 6588–6597.

NABATI, M.; BEHRAD, A. Video captioning using boosted and parallel long short-term memory networks. **Computer Vision and Image Understanding**, v. 190, p. 102840, 2020.

NG, A. Y. Feature selection, L1 vs. L2 regularization, and rotational invariance. *In*: **Proc. of the Twenty-first International Conference on Machine Learning**. Banff, Canada: *[s.n.]*, 2004. p. 78.

NG, H. T.; ZELLE, J. Corpus-based approaches to semantic interpretation in NLP. **AI Magazine**, v. 18, n. 4, p. 45–45, 1997.

NIAN, F.; LI, T.; WANG, Y. *et al.* Learning explicit video attributes from mid-level representation for video captioning. **Computer Vision and Image Understanding**, v. 163, p. 126 – 138, 2017.

OLIVASTRI, S.; SINGH, G.; CUZZOLIN, F. End-to-end video captioning. *In*: **Proc. of the IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)**. Seoul, Korea: *[s.n.]*, 2019. p. 1474–1482.

OTTER, D. W.; MEDINA, J. R.; KALITA, J. K. A survey of the usages of deep learning for natural language processing. **IEEE Transactions on Neural Networks and Learning Systems**, v. 32, n. 2, p. 604–624, 2021.

OURA, S.; MATSUKAWA, T.; SUZUKI, E. Multimodal deep neural network with image sequence features for video captioning. *In*: **Proc. of the International Joint Conference on Neural Networks (IJCNN)**. Rio de Janeiro, Brazil: *[s.n.]*, 2018. p. 1–7.

OZA, P.; PATEL, V. M. C2AE: Class conditioned auto-encoder for open-set recognition. *In*: **Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. *[S.l.: s.n.]*, 2019.

PAN, B.; CAI, H.; HUANG, D.; LEE, K.; GAIDON, A.; ADELI, E.; NIEBLES, J. C. Spatio-temporal graph for video captioning with knowledge distillation. *In*: **Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. *[S.l.: s.n.]*, 2020. p. 10870–10879.

PAN, Y.; YAO, T.; LI, H.; MEI, T. Video captioning with transferred semantic attributes. *In*: **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Honolulu, Hawaii: *[s.n.]*, 2017. p. 984–992.

PAPINENI, K.; ROUKOS, S.; WARD, T.; ZHU, W. BLEU: A method for automatic evaluation of machine translation. *In*: **Proc. of the 40th Annual Meeting on Association for Computational Linguistics**. Philadelphia, USA: *[s.n.]*, 2002. p. 311–318.

PARMAR, J.; CHOUHAN, S.; RAYCHOUDHURY, V. *et al.* Open-world machine learning: Applications, challenges, and opportunities. **ACM Computer Survey**, v. 55, n. 10, p. 1–37, 2023.

PEI, W.; ZHANG, J.; WANG, X.; KE, L.; SHEN, X.; TAI, Y. Memory-attended recurrent network for video captioning. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Long Beach, USA: *[s.n.]*, 2019. p. 8347–8356.

PENG, Y.; WANG, C.; PEI, Y.; LI, Y. Video captioning with global and local text attention. **The Visual Computer**, v. 38, n. 12, p. 4267–4278, 2022.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. *In*: **Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. *[S.l.: s.n.]*, 2014. p. 1532–1543.

PEREZ-MARTIN, J.; BUSTOS, B.; PEREZ, J. Improving video captioning with temporal composition of a visual-syntactic embedding. *In*: **Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)**. *[S.l.: s.n.]*, 2021. p. 3039–3049.

PERLIN, H. A.; LOPES, H. S. Extracting human attributes using a convolutional neural network approach. **Pattern Recognition Letters**, v. 68, p. 250 – 259, 2015.

POUYANFAR, S.; SADIQ, S.; YAN, Y.; TIAN, H.; TAO, Y.; REYES, M. P.; SHYU, M.; CHEN, S.; IYENGAR, SS. A survey on deep learning: Algorithms, techniques, and applications. **ACM Computing Surveys (CSUR)**, v. 51, n. 5, p. 1–36, 2018.

PRATI, A.; SHAN, C.; WANG, K.I. Sensors, vision and networks: From video surveillance to activity recognition and health monitoring. **Journal of Ambient Intelligence and Smart Environments**, v. 11, n. 1, p. 5–22, 2019.

PU, Y.; MIN, M.; GAN, Z.; CARIN, L. Adaptive feature abstraction for translating video to text. *In*: **Proc. of the 32nd AAAI Conference on Artificial Intelligence**. New Orleans, USA: *[s.n.]*, 2018. p. 7284–7291.

QI, M.; WANG, Y.; LI, A. *et al.* Sports video captioning by attentive motion representation based hierarchical recurrent neural networks. *In*: **Proc. of the 1st International Workshop on Multimedia Content Analysis in Sports**. New York, NY, USA: ACM, 2018. p. 77–85.

QI, S.; YANG, L. Video captioning via a symmetric bidirectional decoder. **IET Computer Vision**, v. 15, n. 4, p. 283–296, 2021.

QI, Y.; SACHAN, D.; FELIX, M. *et al.* When and why are pre-trained word embeddings useful for neural machine translation? *In*: **Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. New Orleans, USA: Association for Computational Linguistics, 2018. v. 2, p. 529–535.

RADFORD, A.; KIM, J. w.; HALLACY, C. *et al.* Learning transferable visual models from natural language supervision. *In*: MEILA, M.; ZHANG, T. (Ed.). **Proc. of the 38th International Conference on Machine Learning**. *[S.l.: s.n.]*, 2021. p. 8748–8763.

RASHMI, BS; NAGENDRASWAMY, HS. Video shot boundary detection using block based cumulative approach. **Multimedia Tools and Applications**, Springer, v. 80, p. 641–664, 2021.

RATCLIFF, R. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. **Psychological Review**, v. 97, n. 2, p. 285, 1990.

REIMERS, Nils; GUREVYCH, Iryna. Sentence-BERT: Sentence embeddings using siamese BERT-networks. *In*: **Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. Association for Computational Linguistics, 2019. Disponível em: https://arxiv.org/abs/1908.10084.

ROHRBACH, A.; ROHRBACH, M.; QIU, W. *et al.* Coherent multi-sentence video description with variable level of detail. *In*: **Proc. of the 36th German Conference on Pattern Recognition**. Münster, Germany: *[s.n.]*, 2014. p. 184–195.

ROHRBACH, A.; ROHRBACH, M.; TANDON, N.; SCHIELE, B. A dataset for movie description. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Boston, USA: *[s.n.]*, 2015. p. 3202–3212.

ROHRBACH, A.; TORABI, A.; ROHRBACH, M.; TANDON, N.; PAL, C.; LAROCHELLE, H.; COURVILLE, A.; SCHIELE, B. Movie description. **International Journal of Computer Vision**, v. 123, n. 1, p. 94–120, 2017.

ROHRBACH, M.; QIU, W.; TITOV, I. *et al.* Translating video content to natural language descriptions. *In*: **Proc. of the IEEE International Conference on Computer Vision (ICCV)**. *[S.l.: s.n.]*, 2013.

RUDD, E. M.; JAIN, L. P.; SCHEIRER, W. J.; BOULT, T. E. The extreme value machine. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 40, n. 3, p. 762–768, 2018.

RYU, H.; KANG, S.; KANG, H.; YOO, C. D. Semantic grouping network for video captioning. *In*: **Proc. of the 35th AAAI Conference on Artificial Intelligence**. *[S.l.: s.n.]*, 2021. v. 35, n. 3, p. 2514–2522.

SAH, S.; NGUYEN, T.; PTUCHA, R. Understanding temporal structure for video captioning. **Pattern Analysis and Applications**, v. 23, n. 1, p. 147–159, 2020.

SAHOO, S. P.; ARI, S. On an algorithm for human action recognition. **Expert Systems with Applications**, v. 115, p. 524–534, 2019.

SALEEM, S.; DILAWARI, A.; KHAN, U. G.; IQBAL, R.; WAN, S.; UMER, T. Stateful human-centered visual captioning system to aid video surveillance. **Computers & Electrical Engineering**, v. 78, n. 1, p. 108 – 119, 2019.

SANTOS, M. S.; ABREU, P. H.; JAPKOWICZ, N. *et al.* A unifying view of class overlap and imbalance: Key concepts, multi-view panorama, and open avenues for research. **Information Fusion**, v. 89, p. 228–253, 2023. ISSN 1566-2535.

SCHEIRER, W. J.; Rezende Rocha, A.; SAPKOTA, A. *et al.* Toward open set recognition. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 35, n. 7, p. 1757–1772, 2013.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. **Neural Networks**, v. 61, p. 85 – 117, 2015.

SELJAN, S.; BRKIĆ, M.; VIČIĆ, T. BLEU Evaluation of Machine-Translated English-Croatian Legislation. *In*: **Proc. of the Eighth International Conference on Language Resources and Evaluation (LREC)**. Istanbul, Turkey: European Language Resources Association (ELRA), 2012. p. 2143–2148.

SERRA, J.; SURIS, D.; MIRON, M.; KARATZOGLOU, A. Overcoming catastrophic forgetting with hard attention to the task. *In*: **International Conference on Machine Learning**. Vienna, Austria: *[s.n.]*, 2018. p. 4548–4557.

SHARIF, N.; WHITE, L.; BENNAMOUN, M. *et al.* NNEval: Neural network based evaluation metric for image captioning. *In*: **Proc. of the European Conference on Computer Vision (ECCV)**. *[S.l.: s.n.]*, 2018. p. 37–53.

SHARIF, N.; WHITE, L.; BENNAMOUN, M. *et al.* LCEval: Learned composite metric for caption evaluation. **International Journal of Computer Vision**, v. 127, n. 10, p. 1586–1610, 2019.

SHARIF, N.; WHITE, L.; BENNAMOUN, M. *et al.* WEmbSim: A simple yet effective metric for image captioning. *In*: **Proc. of IEEE Digital Image Computing: Techniques and Applications**. *[S.l.: s.n.]*, 2020. p. 1–8.

SHEN, Z.; LI, J.; SU, Z.; LI, M.; CHEN, Y.; JIANG, Y.; XUE, X. Weakly supervised dense video captioning. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Honolulu, Hawaii: *[s.n.]*, 2017. p. 5159–5167.

SHI, X.; CAI, J.; JOTY, S.; GU, J. Watch it twice: Video captioning with a refocused video encoder. *In*: **Proc. of the 27th ACM International Conference on Multimedia**. New York, NY, USA: ACM, 2019. p. 818–826.

SHI, Y.; YANG, X.; XU, H. *et al.* EMScore: Evaluating video captioning via coarse-grained and fine-grained embedding matching. *In*: **Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. *[S.l.: s.n.]*, 2022. p. 17929–17938.

SHIGETO, Y.; YOSHIKAWA, Y.; LIN, J.; TAKEUCHI, A. Video caption dataset for describing human actions in Japanese. *In*: **Proc. of the 12th Language Resources and Evaluation Conference**. Marseille, France: *[s.n.]*, 2020. p. 4664–4670.

SIGURDSSON, G. A.; VAROL, G.; WANG, X.; FARHADI, A.; LAPTEV, I.; GUPTA, A. Hollywood in homes: Crowdsourcing data collection for activity understanding. *In*: **Proc. of the European Conference on Computer Vision (ECCV)**. Amsterdam: Springer International Publishing, 2016. p. 510–526.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *In*: **Proc. of the IEEE 6th International Conference on Learning Representations (ICLR)**. San Diego, USA: *[s.n.]*, 2015. p. 1–14.

SINGH, T.; VISHWAKARMA, D. K. Human activity recognition in video benchmarks: A survey. *In*: RAWAT, B.; TRIVEDI, A.; MANHAS, S.; KARWAL, V. (Ed.). **Advances in Signal Processing and Communication**. Singapore: Springer, 2019. p. 247–259.

SOBUE, R.; NAKAZAWA, M.; CHAE, Y.; STENGER, B.; YAMASHITA, T.; FUJIYOSHI, H. Cooking video summarization guided by matching with step-by-step recipe photos. *In*: **Proc. of the 16th International Conference on Machine Vision Applications (MVA)**. *[S.l.: s.n.]*, 2019. p. 1–6.

SONG, J.; GAO, L.; GUO, Z.; LIU, W.; ZHANG, D.; SHEN, H. T. Hierarchical LSTM with adjusted temporal attention for video captioning. *In*: **Proc. of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)**. *[S.l.: s.n.]*, 2017. p. 2737–2743.

SONG, J.; GUO, Y.; GAO, L.; LI, X.; HANJALIC, A.; SHEN, H. T. From deterministic to generative: Multimodal stochastic RNNs for video captioning. **IEEE Transactions on Neural Networks and Learning Systems**, v. 30, n. 10, p. 3047–3058, 2019.

SONG, P.; GUO, D.; CHENG, J.; WANG, M. Contextual attention network for emotional video captioning. **IEEE Transactions on Multimedia**, p. 1–11, 2022.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A. *et al.* Dropout: a simple way to prevent neural networks from overfitting. **The Journal of Machine Learning Research**, v. 15, n. 1, p. 1929–1958, 2014.

SUIN, M.; RAJAGOPALAN, A. An efficient framework for dense video captioning. *In*: **Proc. of the 34th AAAI Conference on Artificial Intelligence**. New York, USA: *[s.n.]*, 2020. p. 12039–12046.

SZEGEDY, C.; IOFFE, S.; VANHOUCKE, V. *et al.* Inception-v4, Inception-ResNet and the impact of residual connections on learning. **Proc. of the 31st AAAI Conference on Artificial Intelligence**, v. 31, n. 1, Feb. 2017.

TANG, P.; WANG, H.; LI, Q. Rich visual and language representation with complementary semantics for video captioning. **ACM Transactions on Multimedia Computing, Communications, and Applications**, New York, NY, USA, v. 15, n. 2, jun. 2019.

TORABI, A.; PAL, C.; LAROCHELLE, H.; COURVILLE, A. Using descriptive video services to create a large data source for video annotation research. *In*: **preprint arXiv: 1503.01070**. *[S.l.: s.n.]*, 2015. p. 1–7.

TRAN, D.; BOURDEV, L.; FERGUS, R.; TORRESANI, L.; PALURI, M. Learning spatiotemporal features with 3D convolutional networks. *In*: **Proc. of the IEEE International Conference on Computer Vision (ICCV)**. Santiago, Chile: *[s.n.]*, 2015. p. 4489–4497.

TSAI, J.; HSU, C.; WANG, W.; HUANG, S. Deep learning-based real-time multiple-person action recognition system. **Sensors**, v. 20, n. 17, p. 4758, 2020.

TU, Y.; ZHANG, X.; LIU, B.; YAN, C. Video description with spatial-temporal attention. *In*: **Proc. of the 25th ACM International Conference on Multimedia**. Mountain View, USA: *[s.n.]*, 2017. p. 1014–1022.

TU, Y.; ZHOU, C.; GUO, J. *et al.* Enhancing the alignment between target words and corresponding frames for video captioning. **Pattern Recognition**, v. 111, p. 107702, 2021.

ULLAH, N.; MOHANTA, P. P. Thinking hallucination for video captioning. *In*: **Proc. of the Asian Conference on Computer Vision (ACCV)**. *[S.l.: s.n.]*, 2022. p. 3654–3671.

van de VEN, G. M.; TUYTELAARS, T.; TOLIAS, A. S. Three types of incremental learning. **Nature Machine Intelligence**, Nature Publishing Group, v. 4, n. 12, p. 1185–1197, 2022.

VASWANI, A.; SHAZEER, N.; PARMAR, N. *et al.* Attention is all you need. *In*: **Proc. of the 31th Conference on Neural Information Processing Systems (NeurIPS)**. Long Beach, USA: *[s.n.]*, 2017. p. 1–11.

VEDANTAM, R.; ZITNICK, C. L.; PARIKH, Devi. CIDEr: Consensus-Based Image Description Evaluation. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Boston, USA: *[s.n.]*, 2015. p. 4566–4575.

VENUGOPALAN, S.; ROHRBACH, M.; DONAHUE, J.; MOONEY, R.; DARRELL, T.; SAENKO, K. Sequence to sequence-video to text. *In*: **Proc. of the IEEE International Conference on Computer Vision (ICCV)**. Boston, USA: *[s.n.]*, 2015. p. 4534–4542.

VENUGOPALAN, S.; XU, H.; DONAHUE, J. *et al.* Translating videos to natural language using deep recurrent neural networks. *In*: **Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics**. Denver, USA: *[s.n.]*, 2015. p. 1494–1504.

WANG, B.; MA, L.; ZHANG, W. *et al.* Reconstruction network for video captioning. *In*: **Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. Salt Lake City, USA: *[s.n.]*, 2018. p. 7622–7631.

WANG, B.; MA, L.; ZHANG, W.; JIANG, W.; WANG, J.; LIU, W. Controllable video captioning with POS sequence guidance based on gated fusion network. *In*: **Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)**. Seoul, Korea: *[s.n.]*, 2019. p. 2641–2650.

WANG, D.; SONG, D. Video captioning with semantic information from the knowledge base. *In*: **Proc. of the IEEE International Conference on Big Knowledge (ICBK)**. Hefei, China: *[s.n.]*, 2017. p. 224–229.

WANG, H.; GAO, C.; HAN, Y. Sequence in sequence for video captioning. **Pattern Recognition Letters**, v. 130, n. 1, p. 327 – 334, 2020.

WANG, H.; LIN, G.; HOI, S. C. H.; MIAO, C. Cross-modal graph with meta concepts for video captioning. **IEEE Transactions on Image Processing**, v. 31, p. 5150–5162, 2022.

WANG, Hanli; TANG, Pengjie; LI, Qinyu; CHENG, Meng. Emotion expression with fact transfer for video description. **IEEE Transactions on Multimedia**, IEEE, v. 24, p. 715–727, 2021.

WANG, H.; XU, Y.; HAN, Y. Spotting and aggregating salient regions for video captioning. *In*: **Proc. of the 26th ACM International Conference on Multimedia**. New York, NY, USA: ACM, 2018. p. 1519–1526.

WANG, J.; JIANG, W.; MA, L.; LIU, W.; XU, Y. Bidirectional attentive fusion with context gating for dense video captioning. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Salt Lake City, USA: *[s.n.]*, 2018. p. 7190–7198.

WANG, J.; WANG, W.; HUANG, Y. *et al.* Hierarchical memory modeling for video captioning. *In*: **Proc. of the 26th ACM international conference on Multimedia**. *[S.l.: s.n.]*, 2018. p. 63–71.

WANG, J.; WANG, W.; HUANG, Y. *et al.* M3: Multimodal Memory Modelling for Video Captioning. *In*: **Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. Salt Lake City, USA: *[s.n.]*, 2018. p. 7512–7520.

WANG, S.; YAO, Z.; WANG, R. *et al.* FAIEr: Fidelity and adequacy ensured image caption evaluation. *In*: **Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. *[S.l.: s.n.]*, 2021. p. 14050–14059.

WANG, X.; WU, J.; CHEN, J. *et al.* VaTeX: A large-scale, high-quality multilingual dataset for video-and-language research. *In*: **Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)**. Seoul, Korea: *[s.n.]*, 2019. p. 4581–4591.

WANG, X.; WU, J.; ZHANG, D. *et al.* Learning to compose topic-aware mixture of experts for zero-shot video captioning. *In*: **Proc. of the 33rd AAAI Conference on Artificial Intelligence**. Honolulu, Hawaii: *[s.n.]*, 2019. v. 33, p. 8965–8972.

WANG, Y.; HUANG, G.; YUMING, L. *et al.* MIVCN: Multimodal interaction video captioning network based on semantic association graph. **Applied Intelligence**, Springer, v. 52, n. 5, p. 5241–5260, 2022.

WEI, R.; MI, L.; HU, Y.; CHEN, Z. Exploiting the local temporal information for video captioning. **Journal of Visual Communication and Image Representation**, USA, v. 67, n. C, p. 102751, 2020.

WOJKE, N.; BEWLEY, A.; PAULUS, D. Simple online and real-time tracking with a deep association metric. *In*: **Proc. of the IEEE International Conference on Image Processing (ICIP)**. Beijing, China: *[s.n.]*, 2017. p. 3645–3649.

WOLF, C.; LOMBARDI, E.; MILLE, J. *et al.* Evaluation of video activity localizations integrating quality and quantity measurements. **Computer Vision and Image Understanding**, v. 127, p. 14 – 30, 2014.

WU, B.; NIU, G.; YU, J. *et al.* Towards knowledge-aware video captioning via transitive visual relationship detection. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 32, n. 10, p. 6753–6765, 2022.

WU, X.; LI, G.; CAO, Q. *et al.* Interpretable video captioning via trajectory structured localization. *In*: **Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. Salt Lake City, USA: *[s.n.]*, 2018. p. 6829–6837.

XIAO, H.; SHI, J. Video captioning using hierarchical multi-attention model. *In*: **Proc. of the 2nd International Conference on Advances in Image Processing**. New York, USA: ACM, 2018. p. 96–101.

XIAO, H.; SHI, J. Video captioning with adaptive attention and mixed loss optimization. **IEEE Access**, v. 7, p. 135757–135769, 2019.

XIAO, H.; SHI, J. Video captioning with text-based dynamic attention and step-by-step learning. **Pattern Recognition Letters**, v. 133, n. 1, p. 305 – 312, 2020.

XIAO, H.; SHI, J. Diverse video captioning through latent variable expansion. **Pattern Recognition Letters**, v. 160, p. 19–25, 2022.

XIAO, H.; XU, J.; SHI, J. Exploring diverse and fine-grained caption for video by incorporating convolutional architecture into LSTM-based model. **Pattern Recognition Letters**, v. 129, n. 1, p. 173 – 180, 2020.

XIAO, X.; ZHANG, Y.; FENG, R. *et al.* Video captioning with temporal and region graph convolution network. *In*: **Proc. of the IEEE International Conference on Multimedia and Expo (ICME)**. London, UK: *[s.n.]*, 2020. p. 1–6.

XU, H.; LI, B.; RAMANISHKA, V. *et al.* Joint event detection and description in continuous video streams. *In*: **Proc. of the IEEE Winter Applications of Computer Vision Workshops (WACVW)**. Waikoloa Village, Hawaii: *[s.n.]*, 2019. p. 25–26.

XU, J.; MEI, T.; YAO, T.; RUI, Y. MSR-VTT: A large video description dataset for bridging video and language. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Las Vegas, USA: *[s.n.]*, 2016. p. 5288–5296.

XU, J.; WEI, H.; LI, L. *et al.* Video description model based on temporal-spatial and channel multi-attention mechanisms. **Applied Sciences**, v. 10, n. 12, p. 1–18, 2020.

XU, J.; XU, T.; TIAN, X. *et al.* Context gating with short temporal information for video captioning. *In*: **Proc. of the International Joint Conference on Neural Networks (IJCNN)**. Budapest, Hungary: *[s.n.]*, 2019. p. 1–7.

XU, J.; YAO, T.; ZHANG, Y.; MEI, T. Learning multimodal attention LSTM networks for video captioning. *In*: **Proc. of the 25th ACM International Conference on Multimedia**. New York, NY, USA: *[s.n.]*, 2017. p. 537–545.

XU, N.; LIU, A.; WONG, Y.; ZHANG, Y.; NIE, W.; SU, Y.; KANKANHALLI, M. Dual-stream recurrent neural network for video captioning. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 29, n. 8, p. 2482–2493, 2019.

XU, W.; MIAO, Z.; YU, J. *et al.* Bridging video and text: A two-step polishing transformer for video captioning. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 32, n. 9, p. 6293–6307, 2022.

XU, Y.; YANG, J.; MAO, K. Semantic-filtered soft-split-aware video captioning with audio-augmented feature. **Neurocomputing**, v. 357, n. 1, p. 24 – 35, 2019.

XUE, P.; ZHOU, B. Exploring the spatio-temporal aware graph for video captioning. **IET Computer Vision**, v. 16, n. 5, p. 456–467, 2022.

YAN, C.; TU, Y.; WANG, X. *et al.* STAT: Spatial-temporal attention mechanism for video captioning. **IEEE Transactions on Multimedia**, v. 22, n. 1, p. 229–241, 2020.

YAN, L.; MA, S.; WANG, Q. *et al.* Video captioning using global-local representation. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 32, n. 10, p. 6642–6656, 2022.

YAN, L.; WANG, Q.; CUI, Y. *et al.* GL-RG: Global-local representation granularity for video captioning. *In*: **Proc. of the 31st International Joint Conference on Artificial Intelligence, (IJCAI)**. *[S.l.: s.n.]*, 2022. p. 2769–2775.

YAN, S.; XIONG, Y.; LIN, D. Spatial temporal graph convolutional networks for skeleton-based action recognition. *In*: **Proc. of the 32nd AAAI Conference on Artificial Intelligence**. *[S.l.: s.n.]*, 2018.

YAN, Y.; ZHUANG, N.; NI, B. *et al.* Fine-grained video captioning via graph-based multi-granularity interaction learning. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 44, n. 2, p. 666–683, 2022.

YANG, B.; ZOU, Y.; LIU, F.; ZHANG, C. Non-autoregressive coarse-to-fine video captioning. *In*: **Proc. of the 35th AAAI Conference on Artificial Intelligence**. *[S.l.: s.n.]*, 2021. p. 3119–3127.

YANG, J.; ZHOU, K.; LI, Y.; LIU, Z. Generalized out-of-distribution detection: A survey. **ArXiv**, arXiv:2110.11334, 2021.

YANG, Z.; XU, Y.; WANG, H. *et al.* Multirate multimodal video captioning. *In*: **Proc. of the 25th ACM International Conference on Multimedia**. New York, NY, USA: *[s.n.]*, 2017. p. 1877–1882.

YE, H.; LI, G.; QI, Y. *et al.* Hierarchical modular network for video captioning. *In*: **Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. *[S.l.: s.n.]*, 2022. p. 17939–17948.

YU, J. J. Q.; LAM, A. Y. S.; HILL, D. J.; LI, V. O. K. Delay aware intelligent transient stability assessment system. **IEEE Access**, v. 5, p. 17230–17239, 2017.

Z., Tianyi; K., Varsha; W., Felix *et al.* BERTScore: evaluating text generation with BERT. *In*: **Proc. 8th International Conference on Learning Representations**. *[S.l.: s.n.]*, 2020. p. 1–43.

ZHANG, H.; ZHANG, Y.; ZHONG, B. *et al.* A comprehensive survey of vision-based human action recognition methods. **Sensors**, v. 19, n. 5, p. 1–20, 2019.

ZHANG, J.; PENG, Y. Hierarchical vision-language alignment for video captioning. *In*: **Proc. of the International Conference on Multimedia Modeling**. Thessaloniki, Greece: *[s.n.]*, 2019. p. 42–54.

ZHANG, J.; PENG, Y. Video captioning with object-aware spatio-temporal correlation and aggregation. **IEEE Transactions on Image Processing**, v. 29, n. 1, p. 6209–6222, 2020.

ZHANG, X.; GAO, K.; ZHANG, Y. *et al.* Task-driven dynamic fusion: Reducing ambiguity in video description. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. *[S.l.: s.n.]*, 2017. p. 6250–6258.

ZHANG, Z.; SHI, Y.; YUAN, C. *et al.* Object relational graph with teacher-recommended learning for video captioning. *In*: **Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. *[S.l.: s.n.]*, 2020. p. 13278–13288.

ZHANG, Z.; XU, D.; OUYANG, W. *et al.* Show, tell and summarize: Dense video captioning using visual cue aided sentence summarization. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 30, n. 9, p. 3130–3139, 2020.

ZHANG, Z.; XU, D.; OUYANG, W. *et al.* Dense video captioning using graph-based sentence summarization. **IEEE Transactions on Multimedia**, v. 23, p. 1799–1810, 2021.

ZHAO, B.; LI, X.; LU, X. CAM-RNN: Co-attention model based RNN for video captioning. **IEEE Transactions on Image Processing**, v. 28, n. 11, p. 5552–5565, 2019.

ZHAO, H.; GUO, L.; CHEN, Z. *et al.* Research on video captioning based on multifeature fusion. **Computational Intelligence and Neuroscience**, v. 2022, p. 1–14, 2022.

ZHENG, Q.; WANG, C.; TAO, D. Syntax-aware action targeting for video captioning. *In*: **Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. *[S.l.: s.n.]*, 2020. p. 13096–13105.

ZHENG, Y.; ZHANG, Y.; FENG, R. *et al.* Stacked multimodal attention network for context-aware video captioning. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 32, n. 1, p. 31–42, 2022.

ZHONG, M.; ZHANG, H.; WANG, Y. *et al.* BiTransformer: augmenting semantic context in video captioning via bidirectional decoder. **Machine Vision and Applications**, Springer, v. 33, n. 5, p. 77, 2022.

ZHOU, L.; KALANTIDIS, Y.; CHEN, X. *et al.* Grounded video description. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Long Beach, USA: *[s.n.]*, 2019. p. 6578–6587.

ZHOU, L.; XU, C.; CORSO, J. J. Towards automatic learning of procedures from web instructional videos. *In*: **Proc. of the 32nd AAAI Conference on Artificial Intelligence**. New Orleans, USA: *[s.n.]*, 2018. p. 7590–7598.

ZHOU, L.; ZHOU, Y.; CORSO, J. J. *et al.* End-to-end dense video captioning with masked transformer. *In*: **Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Salt Lake City, EUA: *[s.n.]*, 2018. p. 8739–8748.

ZHU, Y.; JIANG, S. Attention-based densely connected LSTM for video captioning. *In*: **Proc. of the 27th ACM International Conference on Multimedia**. New York, NY, USA: *[s.n.]*, 2019. p. 802–810.

ZHU, Y.; TING, K. M.; ZHOU, Z. New class adaptation via instance generation in one-pass class incremental learning. *In*: **Proc. IEEE International Conference on Data Mining (ICDM)**. New Orleans, USA: *[s.n.]*, 2017. p. 1207–1212.

APPENDIX

APPENDIX A – DETAILED LIST OF THE RELATED VIDEO DESCRIPTION WORKS

Table 18 presents a list of related works analyzed during the period of the thesis.

Table 18 – Summary of video captioning studies present in the literature. S denotes Spatial, T denotes Temporal, R denotes Visual Relations, MM denotes Multimodal, NV denotes Novel Actions, EDL denotes Event Detection and localization, LM denotes Language Model

N.	Author/Year	S	T	R	MM	NV	EDL	LM	Dataset
1	Venugopalan *et al.* (2015b)	AlexNet variant						LSTM	MSVD
2	Venugopalan *et al.* (2015a)	VGG-16	optical flow					LSTM	MSVD, MPII-MD, M-VAD
3	Baraldi *et al.* (2017)	ResNet-50	C3D, LSTM					GRU	MSVD, MPII-MD, M-VAD
4	Gao *et al.* (2017)	Inception-v3	LSTM					LSTM	MSVD, MSR-VTT
5	Krishna *et al.* (2017)		C3D				*	LSTM	ActivityNet Captions
6	Liu *et al.* (2017)	VGG16	LSTM		*			LSTM	MSVD, MPII-MD, MSR-VTT
7	Nian *et al.* (2017)	VGG16, ResNet-152	LSTM		*			LSTM	MSVD, MPII-MD, M-VAD, MSR-VTT
8	Pan *et al.* (2017)	VGG-19	C3D		*			LSTM	MSVD, M-VAD, MPII-MD
9	Shen *et al.* (2017)	VGG-16, ResNet-50	C3D, LSTM		*		*	LSTM	MSR-VTT
10	Song *et al.* (2017)	ResNet-152						LSTM	MSVD, MSR-VTT
11	Tu *et al.* (2017)	Faster R-CNN, GoogLeNet	C3D					LSTM	MSVD, MSR-VTT
12	Wang e Song (2017)	Faster R-CNN, VGG-16	LSTM		*			LSTM	MSVD
13	Xu *et al.* (2017)	GoogLeNet	C3D		*			LSTM	MSVD, MSR-VTT
14	Yang *et al.* (2017)	ResNet-200	C3D		*			GRU	MSR-VTT
15	Zhang *et al.* (2017)	VGG-19, GoogLeNet-bu4k	C3D					LSTM	MSVD, MSR-VTT
16	Chen *et al.* (2018a)	ResNet-152	LSTM					GRU	MSVD, MSR-VTT
17	Chen *et al.* (2018b)	VGG-16	LSTM					LSTM	MSVD
18	Chen *et al.* (2018)	ResNet-152, NasNet	I3D		*			Transformer	MSVD, MSR-VTT
19	Daskalakis *et al.* (2018)	VGG16						LSTM	MSVD
20	Duan *et al.* (2018)		C3D				*	GRU	ActivityNet Captions
21	Li *et al.* (2018)		C3D		*		*	LSTM	ActivityNet Captions
22	Liu *et al.* (2018)	Mask R-CNN, ResNet-101						LSTM	MSVD, MSR-VTT
23	Oura *et al.* (2018)	VGG-16	LSTM		*			NeuralTalk2	MSVD, MSR-VTT
24	Pu *et al.* (2018)		C3D					LSTM	M-VAD, MSVD, MSR-VTT
25	Qi *et al.* (2018)	VGG-16	C3D					LSTM	MSVD, MSR-VTT, SVCDV
26	Wang *et al.* (2018)	Inception-V4						LSTM	MSVD, MSR-VTT
27	Wang *et al.* (2018a)	ResNet-200, GoogLeNet	GRU + VLAD					GRU	MSVD, M-VAD

N.	Author/Year	S	T	R	MM	NV	EDL	LM	Dataset
28	Wang *et al.* (2018b)		C3D				*	LSTM	ActivityNet Captions
29	Wang *et al.* (2018d)	VGG-16, Inception-V3, GoogLeNet	C3D		*			LSTM	MSVD, MSR-VTT
30	Wu *et al.* (2018)	ResNet-152	LSTM						Charades, MSVD
31	Xiao e Shi (2018)	Inception-V3	LSTM		*			LSTM	MSVD
32	Zhou *et al.* (2018b)	ResNet-200	optical flow				*	Transformer	ActivityNet Captions, YouCook2
33	Aafaq *et al.* (2019)	IRv2	C3D		*			GRU	MSVD, MSR-VTT
34	Babariya e Tamaki (2019)	VGG16	LSTM					LSTM	MSVD
35	Barati e Chen (2019)		C3D				*	LSTM	ActivityNet Captions, TACoS-Multilevel
36	Bin *et al.* (2019)	VGG16	LSTM					LSTM	MSVD, MSR-VTT
37	Chen *et al.* (2019)	ResNet-152	LSTM					GRU	MSVD, MSR-VTT
38	Chen *et al.* (2020b)	ResNetXt	ECO		*			LSTM	MSVD, MSR-VTT
39	Chen *et al.* (2019)	Inception-Resnet	C3D		*			LSTM	MSVD, MSR-VTT
40	Chen e Jiang (2019)	Inception-ResNet-V2	C3D		*			LSTM	MSVD, MSR-VTT
41	Dong *et al.* (2019)	ResNet-101	LSTM					LSTM	MSVD, MSR-VTT
42	Du *et al.* (2019)	ResNet-152	C3D, optical flow		*			LSTM	MSVD, MSR-VTT
43	Fang *et al.* (2019)	Inception-V3	C3D					FCN based approach	MSVD, MSR-VTT
44	Francis e Huet (2019)	ResNet-152	I3D					LSTM	MSVD, MSR-VTT
45	Guo *et al.* (2019)	ResNet-152	LSTM		*			LSTM	MSVD, MSR-VTT
46	Hou *et al.* (2019)	Inception-Resnet-V2	C3D		*			POS+ConvCap	ActivityNet Captions, MSVD, MSR-VTT
47	Hu *et al.* (2019)	Faster-RCNN, ResNet-152	C3D					LSTM	Charades, MSVD
48	Jin *et al.* (2019)	Inception-ResNet-V2	I3D		*			LSTM	MSVD, MSR-VTT
49	Lee (2019)	Inception-v4	I3D		*			LSTM	MSR-VTT
50	Lee e Kim (2019)	Inception-V4, ResNet-200	C3D,I3D		*			LSTM	MSVD, MSR-VTT
51	Li *et al.* (2019)	ResNet-152	ResNeXt-101					LSTM	MSR-VTT
52	Li *et al.* (2019)	ResNet, GoogLeNet	C3D					LSTM	MSVD, MSR-VTT
53	Li e Gong (2019)	Inception-ResNet-V2	LSTM					LSTM	MSVD, MSR-VTT
54	Mun *et al.* (2019)		C3D, GRU				*	LSTM	ActivityNet Captions

N.	Author/Year	S	T	R	MM	NV	EDL	LM	Dataset
55	Olivastri *et al.* (2019)	Inception-ResNet-V2	LSTM					LSTM	MSVD, MSR-VTT
56	Pei *et al.* (2019)	ResNet-101	ResNeXt-101					GRU	MSVD, MSR-VTT
57	Saleem *et al.* (2019)	ResNet-152, VGG16	C3D					LSTM	Trecvid 2016, UET- Surveillance
58	Shi *et al.* (2019)	ResNet-152	GRU		*			LSTM	MSVD, MSR-VTT
59	Song *et al.* (2019)	ResNet-152						LSTM	MSVD, MSR-VTT
60	Tang *et al.* (2019)	GoogLeNet, ResNet-101, ResNet-152	LSTM					LSTM	MSVD, MSR-VTT
61	Wang *et al.* (2019)	Inception-ResNet-V2	C3D, I3D					POS+LSTM	MSVD, MSR-VTT
62	Wang *et al.* (2019b)		I3D, LSTM		*	*		LSTM	ActivitiNet Captions, MSR-VTT
63	Xiao e Shi (2019)	Inception-v3	LSTM					LSTM	MSVD, MPII-MD
64	Xu *et al.* (2019)	ResNet-152	3D ResNet (R3D)					GRU	MSVD, MSR-VTT
65	Xu *et al.* (2019)		C3D				*	LSTM	ActivityNet Captions, TACoS-Multilevel
66	Xu *et al.* (2019)	GoogLeNet	LSTM		*			LSTM	MPII-MD, MSVD, MSR-VTT
67	Xu *et al.* (2019)	ResNet-50	3D ResNetXt-101		*			LSTM	MSVD, M-VAD, MSR-VTT
68	Zhao *et al.* (2019)	GoogLeNet, VGG-16	C3D					LSTM	Charades, MSVD, MPII-MD, MSR-VTT
69	Zhang e Peng (2019)	GoogLeNet, MSDN	LSTM						MSVD
70	Zhou *et al.* (2019)	Faster R-CNN, ResNeXt-101, ResNet-101	LSTM					LSTM	ActivityNet-Entities
71	Zhu e Jiang (2019)	VGGNet	C3D					LSTM	MSVD, MSR-VTT
72	Chen *et al.* (2020a)	ResNeXt-101	ECN					GRU	MSVD, MSR-VTT
73	Cherian *et al.* (2020)	Faster R-CNN + ResNet-101	I3D		*			LSTM	MSVD, MSR-VTT
74	Fang *et al.* (2020)	ResNet-152	LSTM					Transformer	V2C
75	Gao *et al.* (2020a)	Resnet-152	C3D					LSTM	LSMDC, MSVD, MSR-VTT
76	Gao *et al.* (2020b)	ResNet-152	C3D		*			GRU	MSVD, MSR-VTT
77	Hemalatha e Sekhar (2020)	ResNet-152	C3D		*			LSTM	MSVD, MSR-VTT
78	Iashin e Rahtu (2020)		I3D		*		*	Transformer	ActivityNet Captions
79	Nabati e Behrad (2020)	ResNet-152	LSTM						MSVD, MSR-VTT

N.	Author/Year	S	T	R	MM	NV	EDL	LM	Dataset
80	Pan *et al.* (2020)	Faster R-CNN, ResNet-101	I3D	STG				Transformer	MSVD, MSR-VTT
81	Sah *et al.* (2020)	ResNet-152	optical flow		*			LSTM	MSVD, M-VAD, MSR-VTT
82	Suin e Rajagopalan (2020)	ResNet-200	LSTM				*	Transformer	ActivityNet Captions
83	Wang *et al.* (2020)	ResNet-200	GRU					GRU	MSVD, M-VAD
84	Wei *et al.* (2020)	ResNet-152	C3D					LSTM	Charades, MSVD, MSR-VTT
85	Xiao *et al.* (2020a)	Inception-v3	LSTM		*			LSTM	MSVD, MSR-VTT
86	Xiao *et al.* (2020b)	Faster R-CNN, Inception-V3	C3D	TGN				GRU	MSVD, MSR-VTT
87	Xiao e Shi (2020)	Inception-v3	C3D		*			LSTM	MSVD, MSR-VTT
88	Xu *et al.* (2020)	Inception-V3	LSTM					LSTM	MSVD, MSR-VTT
89	Yan *et al.* (2022)	Faster R-CNN, VGG-16		KNN-Graph + CNN	*			LSTM	SVN
90	Yan *et al.* (2020)	GoogLeNet, Resnet-152	C3D		*			LSTM	MSVD, MSR-VTT
91	Zhang *et al.* (2020a)	Faster R-CNN (features), IRv2	C3D	GCN				LSTM	MSVD, MSR-VTT, VATEX
92	Zhang e Peng (2020)	ResNet-200	VLAD, ConvGRU	GCN				GRU	MSVD, MSR-VTT
93	Zheng *et al.* (2020)	Faster R-CNN,Inception-ResNet-V2	C3D		*			LSTM	MSVD, MSR-VTT
94	Zhang *et al.* (2020b)		C3D, LSTM				*	LSTM	ActivityNet Captions
95	Ahmed *et al.* (2021)	VGG-16, InceptionV3, Xception, faster R-CNN	I3D					LSTM	MSVD, MSR-VTT
96	Bai *et al.* (2021)	IRV2	I3D	GNN				LSTM	MSVD, MSR-VTT
97	Chen *et al.* (2021)	IRV2	I3D					LSTM	MSVD, MSR-VTT
98	Ji e Wang (2021)	Resnet50						LSTM	MSR-VTT
99	Jin *et al.* (2021)	Inception-ResNet-V2	I3D		*			Transformer	MSVD, MSR-VTT
100	Liu *et al.* (2021)	GoogLeNet, Inception	TCB					LSTM	MSVD, MSR-VTT
101	Perez-Martin *et al.* (2021)	ResNet-152	ECO, d R(2+1)D		*			LSTM	MSVD, MSR-VTT
102	Qi e Yang (2021)	ResNet-152	BiGRU, MH-Att					GRU	MSVD, MSR-VTT
103	Ryu *et al.* (2021)	ResNet-101	ResNext-101,					LSTM	MSVD, MSR-VTT

N.	Author/Year	S	T	R	MM	NV	EDL	LM	Dataset
104	Tu *et al.* (2021)	VGG, ResNet-152			*			LSTM	MSVD, MSR-VTT
105	Yang *et al.* (2021a)	ResNet-101	ResNeXt-101					Transformer	MSVD, MSR-VTT
106	Zhang *et al.* (2021)	ResNet-200			*		*	LSTM+GCN	ActivityNet Captions, YouCook2
107	Ghaderi *et al.* (2022)		Swim video transform		*			Transformer	MSVD, MSR-VTT, Vatex
108	Ji *et al.* (2022)	Inception-V4						LSTM	MSVD, MSR-VTT
109	Lin *et al.* (2022)		I3D		*			Transformer	MSVD, MSR-VTT, VATEX, TVC, YouCook2
110	Li *et al.* (2022b)	IRv2, ResNet-152	optical flow, I3D	GCN	*			LSTM	Charades, MSVD, MSR-VTT
111	Li *et al.* (2022)	IRv2	I3D	LSTG				Transformer	MSVD, MSR-VTT
112	Lu e Gao (2022)	ResNet-152, ResNet-200		GCN	*			LSTM	ActivityNet Captions, Charades
113	Man *et al.* (2022)	ResNet-200, BN-Inception	optical flow		*			Transformer	ActivityNet Captions, YouCook2, and VideoStory
114	Ullah e Mohanta (2022)	ViTL, Faster-RCNN	C3D					LSTM	MSR-VT, MVSD
115	Chen *et al.* (2022)	IRV2	I3D					LSTM	MSVD, MSR-VTT
116	Deng *et al.* (2022)	IRV2	I3D		*			LSTM	MSVD, MSR-VTT
117	Ji *et al.* (2022)	S3D			*			Transformer	YouCook2, ActivityNet Captions
118	Jin *et al.* (2022)	IRV2	I3D		*			Transformer	MSVD, MSR-VTT
119	Li *et al.* (2022a)	ResNet-152	GRU		*			GRU	MSVD, MSR-VTT
120	Peng *et al.* (2022)	ResNeXt	ECO	GCN	*			LSTM	MSVD, MSR-VTT
121	Song *et al.* (2022)	ResNet-101	ResNext-101					LSTM	MSVD, EmVidCap-S, EmVidCap
122	Wang *et al.* (2022)	ResNext	ECHO	GAT	*			LSTM	MSVD, MSR-VTT
123	Wang *et al.* (2022)	ResNeXt	ECO	GCN	*			LSTM	MSVD, MSR-VTT
124	Wu *et al.* (2022)	IRv2	C3D	GCN	*			LSTM	MSVD, MSR-VTT
125	Xiao e Shi (2022)	IRv2	C3D, LSTM		*			LSTM	MSVD, MSR-VTT
126	Xu *et al.* (2022)	ResNeXt	ECO					Transformer	MSVD, MSR-VTT
127	Xue e Zhou (2022)	ResNet-101	TSM-101	GCN				LSTM	MSVD, MSR-VTT
128	Yan *et al.* (2022b)	ResNeXt	Res3D					LSTM	MSVD, MSR-VTT
129	Ye *et al.* (2022)	IRv2	C3D		*			LSTM	MSVD, MSR-VTT
130	Zhao *et al.* (2022)	ResNet	I3D		*			LSTM	MSVD, MSR-VTT
131	Zheng *et al.* (2022)	IRv2, ResNeXt-101	C3D		*			GRU	MSVD, MSR-VTT
132	Zhong *et al.* (2022)	ResNet-152, IRv2	I3D					Transformer	MSVD, MSR-VTT