PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research

BLIDA 1 UNIVERSITY
Faculty of Sciences
Computer Science Department

Master's Thesis
In Computer Science

Option: Software Engineering

Application for contextual images classification

Realised by

- AMEUR El Hachemi
- HAOUI Hamza

Supervised by

- Dr. HIRECHE
- BRAHIM Aimen

Defended on : 24/06/2023 , in front of the jury composed of :
Supervisor : Dr. Hireche
President : Dr. Fareh
Examiner: Ms. Meskaldji

Blida, June 2023

Dedications

I dedicate this master's thesis to my loving parents, thank you for instilling in me a passion for learning and for always believing in my abilities. Your love, sacrifices, and constant encouragement have been instrumental in my journey.

I also dedicate this master’s thesis to my dear friends and colleagues , who have been there for me through the ups and downs of this academic journey. Your knowledge , moral support and shared experiences have made this journey more meaningful and enjoyable.

Furthermore, I would like to dedicate this work to the mentors and professors who have shaped my intellectual growth and inspired me to reach new heights. Your guidance, expertise, and dedication to the field of software engineering have left an indelible mark on my educational journey .

I dedicate this work to madam Hirech Celia for her confidence in us and the knowledge that she gave us during those academic years .

I dedicate this work to our project supervisor Aimen Brahim for his confidence and for helping us realize our ideas and discover our potential.

This master’s thesis is dedicated to the people who have believed in me, supported me, and shared in my excitement and challenges along the way. Thank you all for being an integral part of this achievement.

Haoui Hamza.

Dedications

I Dedicate this work to :

My dear parents, for all their sacrifices, love, support and the heartfelt prayers offered throughout my educational journey.

My brothers redha, amine et akram for their constant encouragement and moral support.

All my friends who accompanied me in this long adventure in search of knowledge and a better future.

My second family it community club and all its members, for their spirit of sharing, for the guidance and for giving me the opportunity to expand my knowledge and skills.

I would also like to express my most sincere thanks to the teachers, students and anyone who has helped directly or indirectly in the preparation of this thesis.

Ameur El Hachemi.

Acknowledgment

First and foremost, we thank ALLAH for giving us the strength, health, and courage to finish this work, which was a cumulation of support, guidance, and encouragement from several people. This has been invaluable throughout this journey.

We would like to thank our master's thesis and project supervisors, Ms. Hireche Celia and Brahim Aimen for their expertise, dedication, and unwavering support. Their insightful feedback, constructive criticism, and continuous guidance have played a pivotal role in shaping this work. We are truly grateful for their mentorship and the opportunities they have provided to us.

Furthermore, we would like to acknowledge our dear parents , families and friends for their unwavering support, patience, and understanding throughout this challenging endeavor. Their encouragement and belief in our abilities have been a constant source of motivation.

Although it is not possible to individually acknowledge everyone who has contributed to our project, please know that your efforts have not gone unnoticed or unappreciated. Thank you all for being part of this significant milestone in our academic and professional journey.

We would also like to thank the members of the jury for having done us the honor of judging and evaluating our work.

Abstract

The goal of this master's thesis is to design, develop, and implement a comprehensive system that can effectively classify images based on their context.

To achieve this objective, we employed two multimodal learning approaches, which enable us to capture and analyze long-term dependencies and contextual information more effectively.

To demonstrate the performance of the proposed methods, experiments were conducted on a custom dataset. The evaluation of the chosen method yielded a classification accuracy of 80%

Key words:

artificial intelligence, image classification, deep learning, contextual image classification, multimodal learning

Résumé

L'objectif de ce mémoire de master est de concevoir, développer et mettre en œuvre un système complet capable de classer efficacement les images en fonction de leur contexte.

Pour atteindre cet objectif, nous avons utilisé deux approches d'apprentissage multimodal, qui nous permettent de capturer et d'analyser plus efficacement les dépendances à long terme et les informations contextuelles.

Pour démontrer la performance des méthodes proposées, des expériences ont été menées sur un jeu de données personnalisé. L'évaluation de la méthode choisie a donné une précision de classification de 80%

Mots clés:

intelligence artificielle, classification d'images, apprentissage profond, classification contextuelle d'images, apprentissage multimod

ملخص

الهدف من أطروحة الماستر هذا هو تصميم وتطوير وتنفيذ نظام شامل يمكنه تصنيف الصور بشكل فعال بناءً على سياقها.

لتحقيق هذا الهدف ، استخدمنا نهجين للتعلم متعدد الوسائط ، مما يمكننا من التقاط وتحليل التبعيات طويلة الأجل والمعلومات السياقية بشكل أكثر فعالية.

لإثبات أداء الطرق المقترحة ، أجريت تجارب على مجموعة بيانات مخصصة. أعطى تقييم الطريقة المختارة دقة تصنيف بلغت 80٪.

الكلمات الدالة:

الذكاء الاصطناعي ، تصنيف الصور ، التعلم العميق ، تصنيف الصور السياقية ، التعلم متعدد الوسائط

TABLE OF CONTENTS

General Introduction....1
1. Project's context....1
2. Problematic....1
3. Objectives....2
4. Document organization....2
Chapter 1: BACKGROUND....3
1. Introduction....3
2. Image classification approaches....3
....
3. Deep Learning....4
3.1. Convolutional neural networks (CNNs)....4
3.2. Recurrent neural networks & Long short memory Term....6
3.3. Sequence modeling....7
3.4. Transformers....7
3.5. Transfer-Learning....10
4. Contextual image classification: from humans to multimodal learning approach....11
5. Evaluation metrics....12
6. Conclusion....12
Chapter 2: Methodology....13
1. Introduction....13
2. Combined Approach....13
3. DataSets....14
4. Image captioning and Chosen solution....15
4.1. Image encoder....15
4.2. Transformer decoder....16
4.3 Training....17
5. Text processing and classification....18
5.1 Training & Evaluation....19
6. CLIP neural network....20
6.1. Vision Transformer....21
6.2. Contrastive learning....22
6.3. Zero-shot classification....23
6.4. Conceptual Captions data-set....25
7. Conclusion....25
Chapter 3: Results & Discussions....26
1. Introduction....26
2. Working environment....26
2.1. Agile methodology for AI project management....26
2.2 Tools and libraries....27
3. Combined model results....28

3.1. Image captioning results..... 28
3.2. Text classification results.....30
3.3. Evaluation..... 30
4. CLIP neural network results.....32
4.1 Evaluation..... 34
5. Final Result..... 35
6. Conclusion..... 36
General Conclusion..... 37
BIBLIOGRAPHY..... 39

TABLE OF FIGURES

Fig 2.1 CNN architecture	6
Fig 2.2 Unrolled Recurrent neural network	6
Fig 2.3 The Transformer - model architecture [1]	8
Fig 2.4 Image explains transfer learning we discussed above [5]	11
Fig 3.1 the main idea	14
Fig 3.2 Image captioning architecture	15
Fig 3.3 image encoding with inception v3	16
Fig 3.4 Result after training	17
Fig 3.5 Stemming example	18
Fig 3.6 Preprocessing	19
Fig 3.7 Constractive pre-training for clip neural network	20
Fig 3.8 ViT Model overview [8]	21
Fig 3.9 Contrastive Learning of Visual Representations [23]	23
Fig 3.10 Example of training data	23
Fig 3.11 Example of test data of unseen classes	24
Tab 1 Image captioning model metrics	29
Fig 5.1 Image from our test set	29
Tab 2 Calculating bleu and rouge for Fig 5.1	29
Tab 3 Text classification model metrics	30
Fig 5.2 Combined Model prediction 1	31
Fig 5.3 Combined Model prediction 1	31
Fig 5.4 Combined Model prediction test on unseen_image	32
Fig 5.5 Using clip for image classification	33
Fig 5.6 CLIP accuracy on the custom dataset	33
Fig 5.7 CLIP image nb1 prediction on testing set	34
Fig 5.8 CLIP image nb2 prediction on testing set	34
Fig 5.9 CLIP prediction on an unseen image	35
Fig 5.10 Classified images based on context after upload 01	35
Fig 5.11 Classified images based on context after upload 02	36

General Introduction

1. Project's context

The field of computer vision has rapidly grown, with many applications across different industries. One of the fundamental tasks in computer vision is image classification, which has been significantly improved with the use of deep learning techniques in recent years. Even with these improvements the current image classification methods still face accuracy limitations in situations where contextual information is crucial and the images contains much information.

Contextual image classification has the potential to revolutionize computer vision if it can consider the context in which an image was taken with more accurate results. we can classify better if we can understand the meaning of the image and its relationship to the world around it. Contextual information can be incorporated into image classification through different approaches, such as multi-scale analysis, spatial attention mechanisms, and graph-based representations. However, these approaches have limitations, including the need for significant human intervention, the difficulty of obtaining contextual information for some types of images, and the lack of robustness of some contextual models in complex environments.

In our world we face numerous examples of image classification , but the most relevant to the issue that we aim to address is Google travel image classifier , that classifies the hotel images taken by the visitors based on context ex :Bedroom,Food,Interior,Exterior and Entertainment .

google also classifies the user comment with photos based on the context of the comment to help their users to make the best possible choice.

In this master's thesis, we aim to solve the icosnet problem which consists in classifying images according to their context. We explore the potential of contextual image classification by combining text and images using multi-modal learning. This model has shown impressive performance in a variety of tasks .

2. Problematic

Ibox is a secure storage solution that allows online file sharing and synchronization. Mainly used by companies as a storage service to collaborate freely and efficiently between their teams, regardless of the distance.

Ibox offers its customers the ability to back up their files without limit. All data is hosted in Algeria at Icosnet Datacenters .

It is a very powerful collaboration tool, with a very useful graphical interface to share files with your collaborators, your customers and your suppliers.

Since icosnet Ibox is file storage and share solution the main challenge is that we can never know exactly what the client will upload to the storage service so we had to deal to train the model in way that it can also classify unseen images that are not close to the images that we had during the training, also we had to find a way to know exactly what are the different actors in the image and the relation between them to get the ability to extract the context .

3. Objectives

The contribution of this study is to provide a comprehensive evaluation of the potential of the multimodal deep learning for contextual image classification, and to identify opportunities for future research and development in this area. By addressing the limitations of existing image classification methods and exploring the potential of contextual information and the multimodal deep learning, this study has the potential to advance the field of computer vision and contribute to the development of more accurate and efficient image classification systems. We aimed to improve contextual image classification, by leveraging both textual and visual information to learn more robust and generalizable representations. However, these models are still relatively new and require large amounts of pre-training data to achieve state-of-the-art performance.

4. Document organization

This document is organized as follows :

Chapter 1 provides a comprehensive review of existing literature and relevant studies related to the research topic.

Followed by Chapter 2 where we described the research methodology, employing two multimodal approaches as our solution for the problem.

In Chapter 3, the findings and results are presented using tables, interpreted in the context of the research objective and discussed.

And finally a general conclusion that includes a summary of the whole process and our perspective on it.

Chapter 1: BACKGROUND

1. Introduction

Image classification is the task of categorizing an image into one or more predefined classes or categories. The goal of image classification is to assign an input image to the correct class ,and this is done by training a model on a set of labeled images, where each image is associated with a label or class. Once trained, the model can be used to classify new images.

In recent years, there has been a growing interest in combining approaches to improve the performance of image classification, many researchers have explored the use of attention mechanisms in CNNs to capture contextual information, while others have used graph-based models or spatial features to represent the context of the scene.

2. Image classification approaches

Image classification refers to the task of extracting information classes from an image, or as a some of image classification technique process of sorting pixels into a finite number of individual classes, or categories of data, based on their spectral response (the measured brightness of a pixel across the image bands, as reflected by the pixel's spectral signature) Image classification involves comparing an object to predefined patterns in order to categorize it appropriately. It is a crucial and challenging task in various application domains.
Depending on the interaction between the analyst and the computer during classification. There are several image classification methods of which the two main are supervised and unsupervised learning. [7]

- **Unsupervised learning**

Unsupervised learning is to regroup a set of images, with the aim that a same cluster contains the most similar images and different clusters contain the most different images.
The algorithm would analyze the images and look for patterns and similarities between them. It would then group together images that have similar patterns and assign them to clusters.

- **Supervised learning**

Supervised classification refers to methods where the classes are known. The model is, in this case, trained to predict the class of newly unlabeled images.

Firstly, a collection of annotated photos for supervised classification is needed. This collection or dataset is then divided into a training and a testing set. The training set would be used to train the machine learning model while the testing set would be used to evaluate the model's accuracy .

3. Deep Learning

Over the past few years, methodologies for extracting information from images for machine understanding have made great progress. During this time, deep learning methods that make use of convolution operations have become a key method of image feature extraction, and often replacing traditional computer vision feature extraction methods. Currently deep learning methods have produced state of the art results for the tasks of object recognition and object recognition. Similar to their adoption for computer vision related tasks, deep learning approaches have also gained popularity in the Natural Language Processing (NLP) domains due to the ability of Recurrent Neural Networks (RNNs) to learn text sequences accurately.

3.1. Convolutional neural networks (CNNs)

From the beginning, Neural Networks were designed to imitate the workings of the human brain as closely as possible. Convolutional Neural Networks (CNNs) specifically contribute to this by working with the visual sensory organs of living beings to recognize various types of objects, including digits, shapes, and faces. Convolutional neural networks are a type of artificial neural network that are designed to process image data by learning and extracting relevant features. This approach has been highly effective in image recognition and classification tasks, enabling the development of applications such as self-driving cars, medical image analysis, and robotics. [15 , 17]

- **CNN Architecture**

CNNs are composed of three types of layers. convolutional layers, pooling layers and fully-connected layers [24 , 15].

Convolutional layers : convolution layer is the most important layer in the CNN, it multiplies a small matrix of weights, called kernel or filter, in small areas of the input image called receptive fields. This operation produces a feature map, which represents the

presence of specific features in the input image. Each element in the feature map corresponds to a specific receptive field in the input image. The value of the element is calculated as the dot product between the kernel and the values in the corresponding receptive field. By sliding the kernel across the entire input image, the convolution layer produces a new feature map with filtered information that captures relevant features in the input image.

ReLu : standing for Rectified linear unit, it is a widely used activation function , the ReLu function returns the input value if it is positive or zero; otherwise returns zero .

A(x) = max(0,x)

The main reason to the use of ReLu function in Cnn because it is a very simple calculation and also it has a derivative of 0 or 1 depending on whether its input is negative or not , as consequence, the usage of ReLU helps to prevent the exponential growth in the computation required to operate the neural network .

Pooling Layers : Pooling is an important step to further reduce the dimensions of feature maps, keeping only the important features while also reducing the spatial invariance. This in turn reduces the number of learnable features for the model and helps to resolve the problem of overfitting. Pooling allows CNN to incorporate all the different dimensions of an image so that it successfully recognises the given object even if its shape is skewed or is present at a different angle. There are various types of pooling like max pooling, average pooling, stochastic pooling, spatial pyramid pooling.Out of them most popular is max pooling.

Fully connected layers : The fully-connected layers have the same function as those in standard Artificial Neural Networks (we can just say neural networks). Their goal is to use the activation values from the previous layer to generate a set of scores for each possible class, which can then be utilized to classify the input data.

Here's an example of CNN typical architecture that was discussed above [*Fig 2.1*].

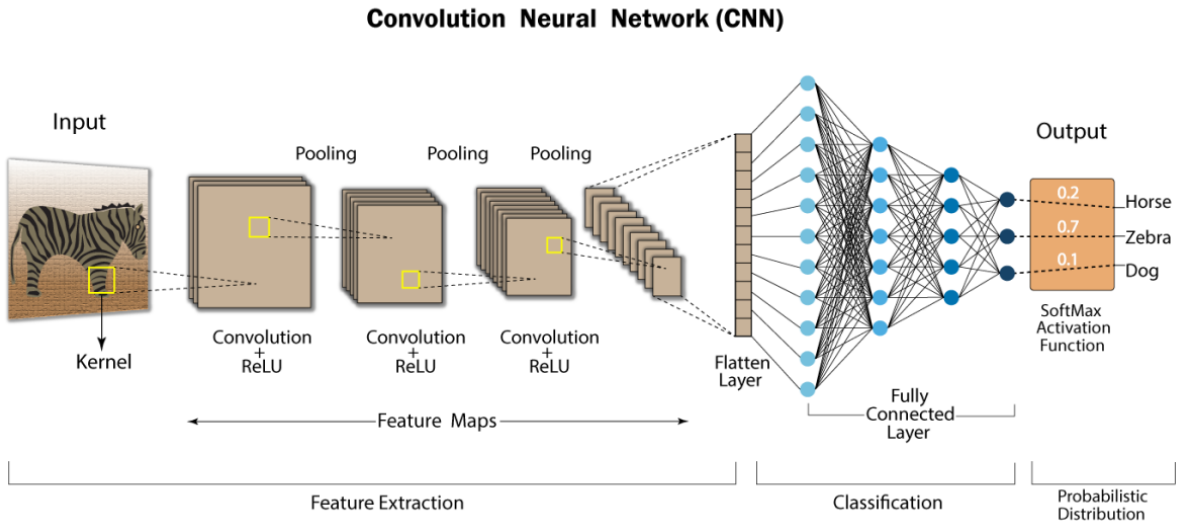

Fig 2.1 CNN architecture [25]

- **CNN Models**

Several CNN models have been developed over time, some of the most notable ones include AlexNet, GoogLeNet, VGG and ResNet, [16 , 24]. These models are, often, used in transfer-learning.

3.2. Recurrent neural networks & Long short memory Term

RNN architecture consists of a series of recurrent layers, where each layer contains a set of hidden units that process the input and maintain a hidden state. The hidden state of the current time step is then used as input to the next time step, along with the input for that time step. This process allows the network to maintain a memory of previous inputs and use that information to make predictions about the current input. [Fig 2.2] where , A : hidden units , Xi : input value , hi : hidden state

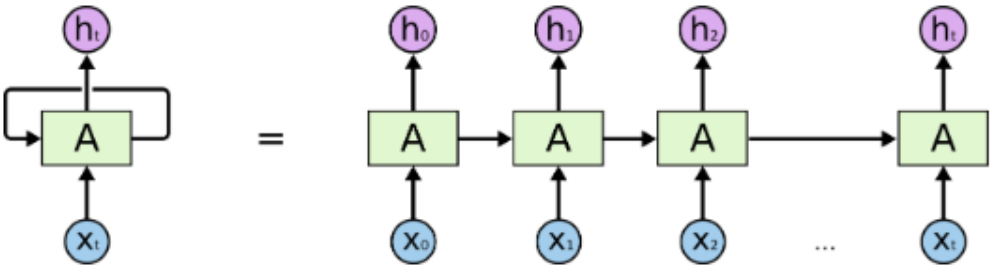

Fig 2.2 Unrolled Recurrent neural network [26]

RNNs are a type of neural network mainly used for NLP tasks, that they can be applied to computer vision tasks, However, their primary strength lies in the ability to model sequential data (text or speech) , by taking into account the context in which word appears.

Regarding capturing long-term dependencies , is one of the challenges that RNNs struggle with , due to a problem called vanishing gradients, where the gradients used to update the weights in the network become very small and effectively disappear over time. As a result, RNNs will have difficulties to remember information from earlier time steps, which can limit their effectiveness for some applications.

To overcome this challenge, Long Short-Term Memory (LSTM) was developed. LSTMs are variants of RNNs That use a gating mechanism to selectively forget or remember information from earlier time steps, allowing them to maintain long-term dependencies more effectively. [17 , 10]

3.3. Sequence modeling

Each text can be split up into a sequence of characters or a sequence of words , Sequence modeling focuses on analyzing and predicting patterns in ordered sets of data (sequences) .

Sequences can be found in a wide variety of applications [18] , including natural language processing, speech recognition, video analysis, and financial forecasting.

In order to model sequences there is some criteria that we need to check [12] :

- that the model can handle sequences of any given length
- the ability to track and learn long term dependencies in the data , the model needs to handle those dependencies that may occur at times that are very distant from each other
- sequences are all about order , it's needed to know how current input depends on the previous and the next one , so a good modal should maintain information about order .
- In order to process information effectively , the model needs to apply the same set of weights at different time steps in the sequence and always result in a good prediction , this is called parameters sharing.

3.4. Transformers

Transformers are a type of neural network architecture that has been widely used in natural language processing tasks, such as language translation and text generation. Unlike other neural network models that process sequential data, such recurrent neural networks, transformers process entire input sequences simultaneously.

While transformers were originally developed for natural language processing tasks, they have recently shown promise in the field of computer vision [22]. In particular, transformer-based models have achieved state-of-the-art performance on image classification, object detection, and segmentation tasks.

The key concept of the transformer architecture is 'attention mechanism' which allows the model to focus on different parts of the input sequence when making predictions. This mechanism works by computing a weighted sum of the input sequence, where the weights are learned based on the relevance of each input element to the prediction task [1].

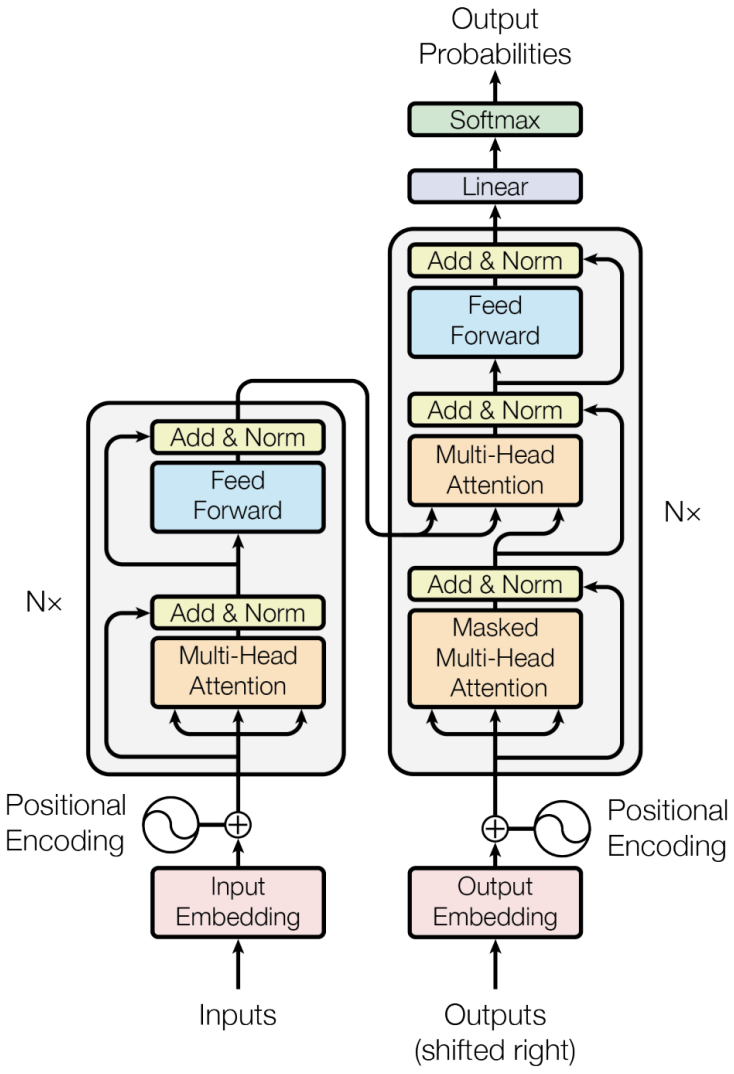

Fig 2.3 The Transformer - model architecture [1]

- **Inputs embeddings**

In transformers, inputs are embedded into a high-dimensional vector space using an embedding matrix; each row of the matrix corresponds to a specific token in the vocabulary.
This process captures the relationships between the different tokens in the input sequence; this allows the transformer model to capture dependencies between elements of the sequence. The input embeddings are then passed to the encoder [1] .

- **Encoder**

The encoder is a component in a transformers architecture that accepts a sequence of input tokens and creates a sequence of encoded representations of those tokens, the encoder is made up of numerous layers of self-attention and feedforward neural networks. [22]
The output of the encoder is a sequence of vectors, each vector representing a different aspect of the input tokens and the relationships between them . These encoded representations can be used by a decoder component to generate the next word in the sequence .

- **Decoder**

A decoder is a component that accepts an encoded input sequence and produces a sequence of output tokens. The decoder is often made of numerous layers of self-attention and feedforward neural networks [1].
The decoder is often autoregressive, which means that it generates each output token based on the previous tokens it has generated. In other words, the decoder generates one token at a time and uses its prior guesses as input for the next prediction.
The attention mechanism is used by the decoder to focus on relevant information of the encoded input sequence,allowing the decoder to consider the relationship between the input and output sequences [22] .

- **Transformers in computer vision**

Transformers have been a breakthrough architecture in the field of natural language processing, and they are now being increasingly used in computer vision as well. The transformer architecture is a type of neural network that is designed to process sequential data, which makes it a natural fit for tasks such as image and video analysis that involve processing of large amounts of sequential data. Transformers in computer vision have been used for a variety of tasks, including image classification, object detection, semantic segmentation, and video analysis. One of the key advantages of transformers is their ability to capture long-range dependencies in data, which is critical for understanding complex visual

patterns. Additionally, transformers have the ability to attend to multiple regions of an image simultaneously, allowing them to efficiently process large images. Recent research has shown that transformers can outperform traditional convolutional neural networks in certain computer vision tasks, indicating their potential to revolutionize the field. As such, transformers are an exciting area of research in computer vision, and are expected to play an increasingly important role in the development of advanced computer vision systems.

3.5. Transfer-Learning

Transfer learning as a new machine learning paradigm has gained increasing attention lately. Large deep models that can be trained on an abundance of labeled data have regularly shown to be the best approach for classifying images.

Unfortunately, there are numerous situations in the actual world where the need for a lot of training data cannot be satisfied in order to achieve the optimal performance [5] . In these scenarios transfer learning can be one of the solutions that helps to improve performance .

The technique of employing a pre-trained neural network that has already been trained on a sizable dataset, such as ImageNet, as a starting point for a new image classification problem is known as transfer learning in the field of image classification. Transfer learning involves using the previously trained network as a feature extractor, where the learned representations of the original network are used to extract features from the new images. These features are then fed into a new classifier to make predictions on the new dataset, as opposed to training a new neural network from scratch on the new dataset.

Due to the pre-trained network's extensive prior knowledge, this method can considerably improve the training process and performance of machine learning models by leveraging pre-trained models to reduce training time, improve generalization, require fewer data, adapt to new domains and achieve better performance compared to starting from scratch.

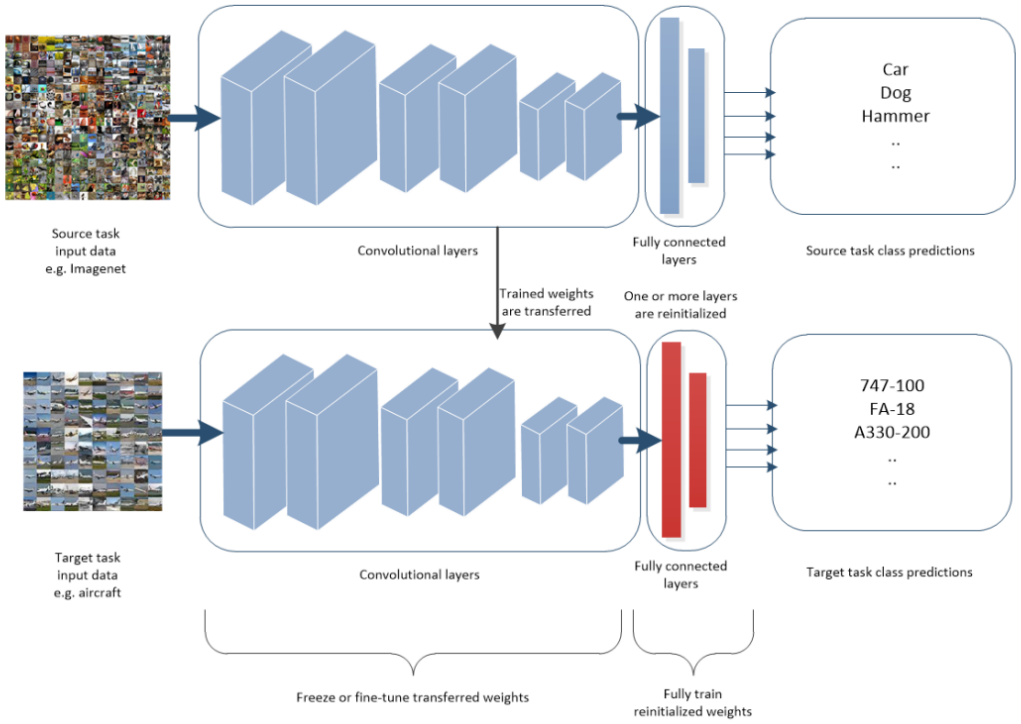

Fig 2.4 Image explains transfer learning we discussed above [5]

A lower learning rate and momentum are better for fine-tuning when the source and target datasets are similar. This may not be necessary when the target dataset is very large. [2]

4. Contextual image classification: from humans to multimodal learning approach

Similar to language processing, single word may have multiple meanings unless the context is provided or the word is on a sentence , and for images if we want to know the context of an image we should be focusing on the relationship between all the details of the image.

Contextual image classification focuses mainly on categorizing images based on their visual content and their relationship with the surrounding environment.

Contextual image classification takes into account the semantic relationship between elements within the image such as the location, the relationships between objects, and the scene's overall context; this should bring more accurate results than traditional image classification. We are discussing the full idea at the beginning of the next chapter .

5. Evaluation metrics

Evaluating the quality of an automatically generated caption is a difficult and subjective task , complicated because one image can have multiple true captions and the generated caption does not have to be fluent, it just needs to refer properly to the content of the image . [21] [3]
The best way to measure the quality of the generated caption is still carefully designing a human evaluation campaign in which multiple users score the produced sentences. [22]
To measure the performance of a machine learning algorithm , we see often the use of Accuracy , Precision and recall

- **Accuracy** : It represents the ratio of correct predictions to the total number of predictions.

Accuracy = (TP + TN) / (TP + FP + TN + FN)

- **Precision** : It is the ratio of the correct positive predictions to the total number of positive predictions

Precision = TP / (TP + FP)

- **Recall** : Recall calculates the ratio of predicted positives to the total number of positive labels.

Recall = TP / (TP + FN)

where :
TP (true positive) : is when the label is positive and our predicted value is positive
TN (true negative) : similar to True Positive, the only difference being the label and predicted value are both negative
FP (false positive) : is when the label is negative but our model's prediction is positive.
FN (false negative) : is when the label is positive but the predicted value is negative

6. Conclusion

In this chapter we explained some of the background study, starting by image classification and its approaches, deep learning methods, transformer and transfer learning, finalizing it by contextual image classification. This chapter was a starting point to understand the research problem and the relevant concepts and techniques. It aimed to provide a comprehensive overview of the key topics related to the research area.

Chapter 2: Methodology

1. Introduction

As humans, when encountering a new image, the first thing to do is to start by closely examining it. Next comes speculating and wondering, “What do I believe I see? What may be happening? Why, in my opinion, was this picture produced? What visual hints brought me to these theories?”

What if, a classification model follows a human brain behavior while trying to discover the context of an image.When it comes to teaching kids, Multimodal learning suggests that a number of our senses (visual, auditory, kinaesthetic) are being engaged during the learning process.

By analogy, in AI, multimodal learning is about building a modal by combining different types of models (NLP, computer vision, speech recognition) and multiple types of data(image text video, audio). Multimodal learning is important for many real world problems that involve complex data, where a single input is not enough .

2. Combined Approach

In our case, following the same human behavior by combining text and images can be effective while trying to understand and find the full content of the image and the relations between those parts. For these reasons, and to do contextual image classification we decided to combine both text and images :

Firstly, the captured image is used to answer all the questions. Then, the context is extracted from the obtained caption from the input image. This process is illustrated in figure 3.1.

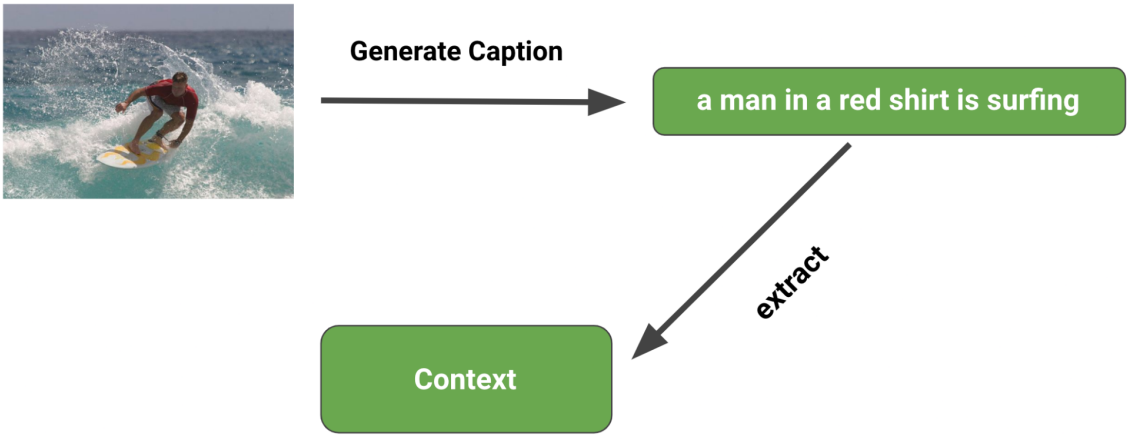

Fig 3.1 The main idea

To do so, multiple techniques are involved such as convolutional neural networks (CNNs) for image processing, Transformers for sequence processing and attention mechanisms to focus on important information across different modalities.

3. DataSets

In order to process image captionning we have chosen to train our model by combining text and images , Our dataset is then constituted by (image,text) pairs.

- **Flickr8k dataset**

Flickr8k is a dataSet that contains more than 8000 pairs of images and their clear descriptions of the salient entities and events descriptions , each image is followed with 5 captions which makes 40000 training pairs [6].

To ensure a variety of scenes and situations, Flickr 8k Dataset photos and captions were manually selected

- **News_category dataset**

We used News_category dataset [14] to help us categorize captions figure 3.1

It represents one of the biggest News datasets and can serve as a benchmark for a variety of computational linguistic tasks. It contains around 210k News headlines from 2012 to 2022 from HuffPost. and 42 News categories such as “entertainment, politics, sports …” .

4. Image captioning and Chosen solution

Since the apparition of object detection, image captioning has been a challenging task in the field of computer vision and natural language processing.

The current state of the art in image captioning is a combination of visual features and language models. Mostly, CNN are used to extract the visual features and Recurrent Neural Networks (RNNs) or Transformer-based models to generate the captions .

This approach has shown a lot of improvement in captioning performance, especially with the use of pre-training techniques like transfer learning and fine-tuning .

One of the most famous models for image captioning is “show and tell model“ [19] ,

that combines CNN’s and LSTM’s , this model was able to achieve state-of-the-art performance on various benchmark datasets such as COCO [13] and Flickr30k [20] .

We kept the same model as the one used in “show and tell model“ [19] but with the use of the transformer model [2.6] instead of LSTM due to its ability to capture long term dependencies and contextual information .

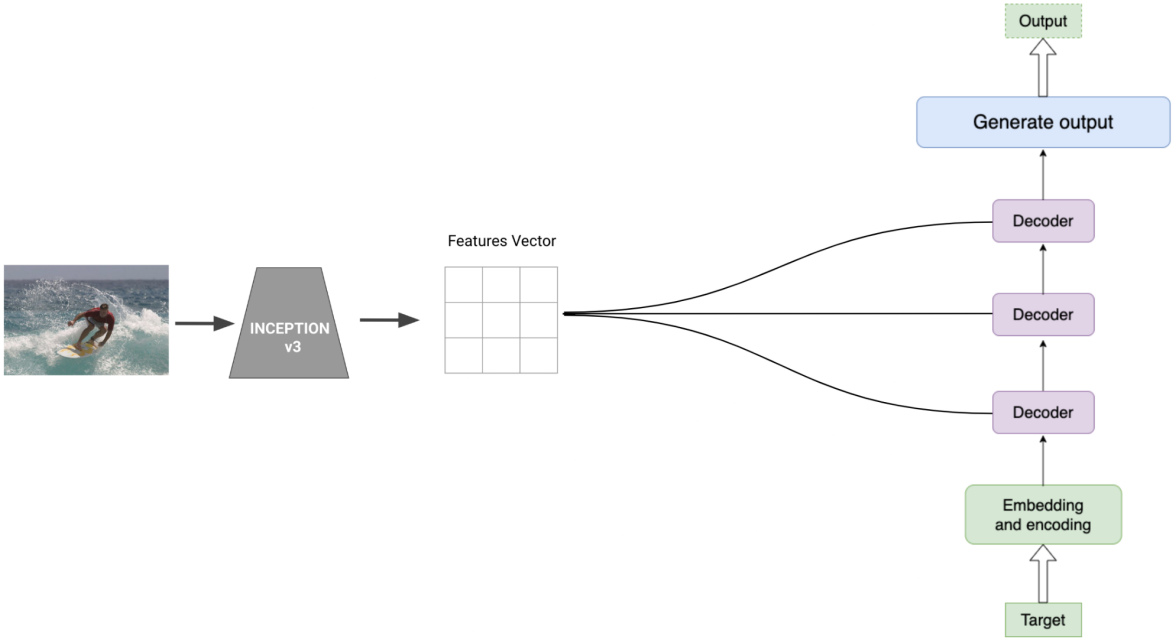

Fig 3.2 Image captioning architecture

4.1. Image encoder

we used the pretrained model Inception V3 that was trained on ImageNet dataset with an accuracy of 78% , and we took all the knowledge of this modal and apply it to our image

captioning problem , after removing the last fully connected layer because we do not want to do image classification in this case .

Before passing the image to inception v3 model we need to process the image with the following steps :

- Read the image and load it in tensor format
- Resize the image into the optimal shape. Indeed, before feeding the image to CNN's it is necessary to resize it to a specific shape. For inception v3 it is 299x299 and has a benefit that helps save a lot of computational power .
- Normalize the image by transforming the pixels values to a standardized range , 0 and 1 in our case , we can do this by dividing the value of each pixel by 255 , normalization can help to reduce the effect of lighting conditions and color difference in the image which can improve the accuracy

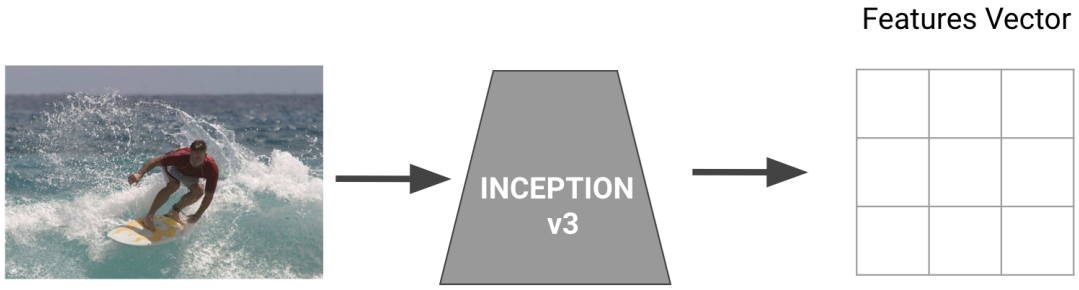

Fig 3.3 Image encoding with inception v3

4.2. Transformer decoder

The output of the encoder is a sequence of contextualized embeddings, which are vector representations of the input visual features the context and relationships between them might look something like this :

[embedding1, embedding2, embedding3] .

Each embedding is a vector of dimensionality 512 that represents a contextualized representation of one of the input visual features .

The decoder will take the sequence of contextualized embeddings generated by the encoder and process them via the self attention and the feed-forward network knowing that the decoder is composed of multiple identical layers each layer contains two sub-layers:

- **Multi-head self-attention mechanism** : this allows the encoder to attend to different parts of the input sequence and the parts of the image that are most relevant for each word in the caption .
- **Position-wise fully connected feed-forward network** :it is applied to each position in the sequence independently. This allows the encoder to capture non-linear relationships between the visual features.

in addition it will have the attention mechanism and a language modal to generate the sequence of words by generating the probability of distribution over the possible next words in the caption at each step then selects the word with the highest probability , this process is repeated until we generate the end-of-sequence token, that indicates that the caption is completed .

4.3 Training

The model was trained with a combination of supervised learning and teacher forcing. The true word that should be predicted is always passed at each time step. This will allow us to predict the next word based on the true word that should be predicted and avoid getting penalized for a wrong prediction that we've made before.

The loss function is typically a cross-entropy loss between the predicted and ground-truth captions .

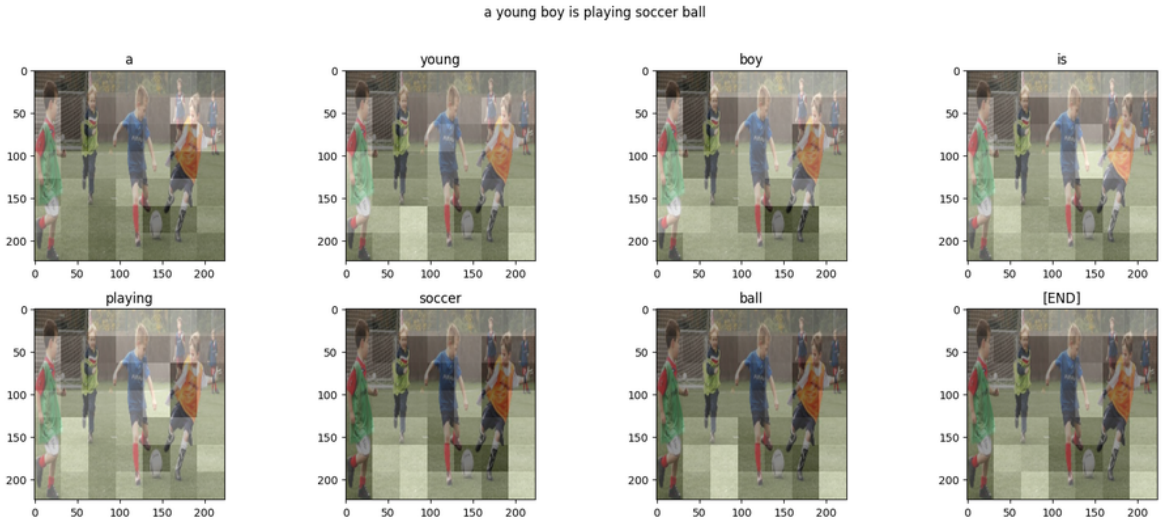

Fig 3.4 Result after training

5. Text processing and classification

Text classification, also known as text categorization, is the process of assigning predefined categories or labels to a given set of textual data.

We employed this method [fig 3.2] to categorize our captions [fig 3.4]. This helps us classify each caption into its corresponding category.

Before applying text classification, a preprocessing is needed and includes :

- Lower case all the captions
- Remove all punctuation and useless spaces
- Add [START] and [END] special tokens , then process the text vectorisation by converting all the caption content to a numerical representation that can be used in machine learning .

After loading the news_categories dataset [3.2.2] ,

- **Normalization**

Normalization involves transforming words to a standard or canonical form , it includes : converting words to lowercase to reduce complexity of the feature space, removing accents from accented characters , removing non alphabetic characters …

- **Stemming**

Stemming is the process of reducing a word to its root form, by removing prefixes and suffixes. example : the word “eating” will be stemmed to “eat” .

This helps reduce the number of unique words in a text corpus, it will be easier to group similar words together.

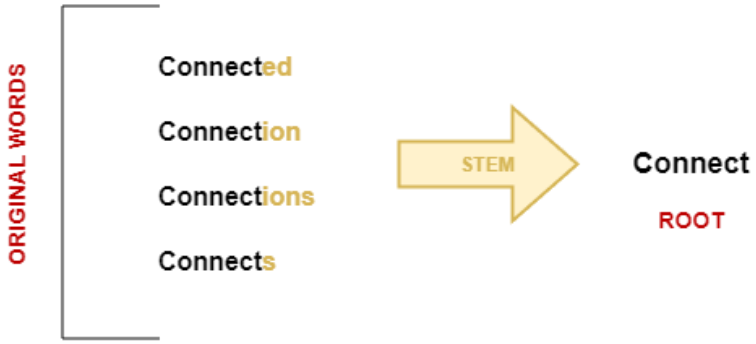

Fig 3.5 Stemming example

- **Removing stopwords**

This step involves removing commonly occurring words in a specific text corpus, such as "the," "in," "a," etc., which are also known as stop words. By removing these stop words, we can reduce the dimensionality of the text data and improve the efficiency and effectiveness of the text classification.

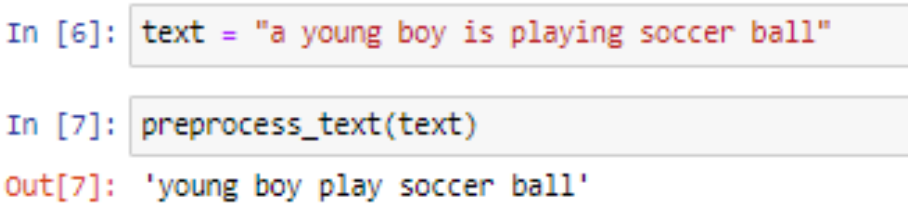

Fig 3.6 Preprocessing

Fig 3.6 represents an input / output example of the text preprocessing.

- **Count vectorization**

To convert the text data into numerical format , we used count vectorization . This involves creating a sparse matrix where each row corresponds to a document in the dataset and each column corresponds to a unique word in the corpus. The values in the matrix represent the frequency of each word in each document. This matrix can then be used as input to a machine learning algorithm.

5.1 Training & Evaluation

The model was trained using logistic regression which is a popular machine learning algorithm that is widely used for binary classification tasks. It can also be extended to handle multi-class classification problems, as in our case. The algorithm was implemented through the use of Scikit-learn's LogisticRegression class, where the fit method was applied to train the model using the training data vectors. Subsequently, the predict method was applied to generate predictions on the testing set, and the accuracy of the model was evaluated using the accuracy_score function from Scikit-learn's metrics module.

6. CLIP neural network

In 2021 we saw the introduction of CLIP neural network (Contrastive Language-Image Pre-Training) which is a neural network architecture that can be effective at performing both image and text based tasks . [11]

It takes both images and text pairs and trains them on to output embedding vectors that are close to each other in order to make them speak the same language ,also the model was trained on full sentences instead of single classes like car, dog, etc. The intuition is that when trained on whole sentences, the model can learn a lot more things and finds some pattern between images and texts. They also show that when this model is trained on a huge dataset of images and their corresponding texts, it can also act as a classifier too.

The model will be trained with two modules. One to encode textual data and the other to encode images data..

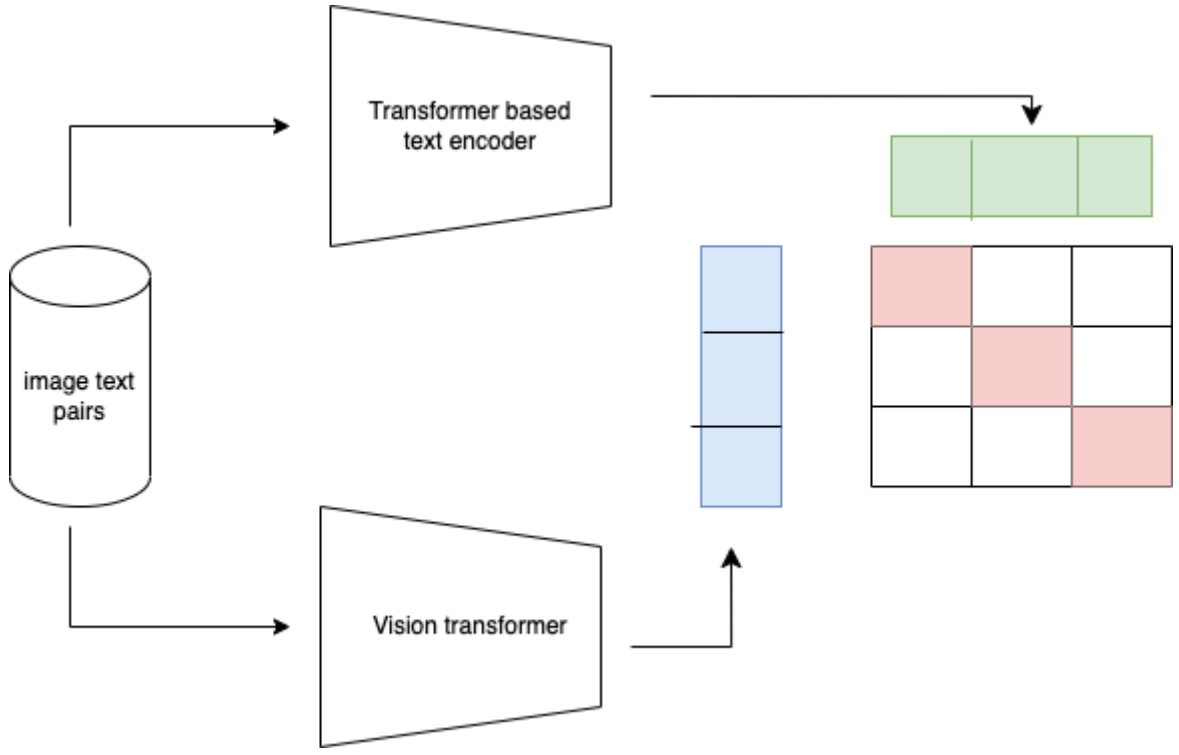

Fig 3.7 Constractive pre-training for clip neural network

6.1. Vision Transformer

The Vision Transformer (ViT) is a type of transformer architecture designed for computer vision tasks. It uses the same self-attention mechanism used in transformers , to images allowing it to capture spatial relationships between different parts of an image.

The vision transformer takes in an image as input and to understand it, VIT divides this image into a grid of patches, where each patch is mapped to a vector using a linear projection. This vector representation is called the "patch embedding" where it captures important information about the patch contents.

These embeddings are then passed into a self-attention layer, where the model attends to various regions of the image and learns spatial relationships between them. Following the self-attention layer, the embeddings are processed through a series of Multi-Layer Perceptrons (MLPs), with residual connections to preserve the original information. This process is repeated over multiple layers of self-attention and MLPs, ultimately producing a final classification output.

Compared to other computer vision architectures like convolutional neural networks (CNNs), ViT has shown promising results on image classification tasks with relatively few parameters, and it can be fine-tuned on downstream tasks with small amounts of labeled data.

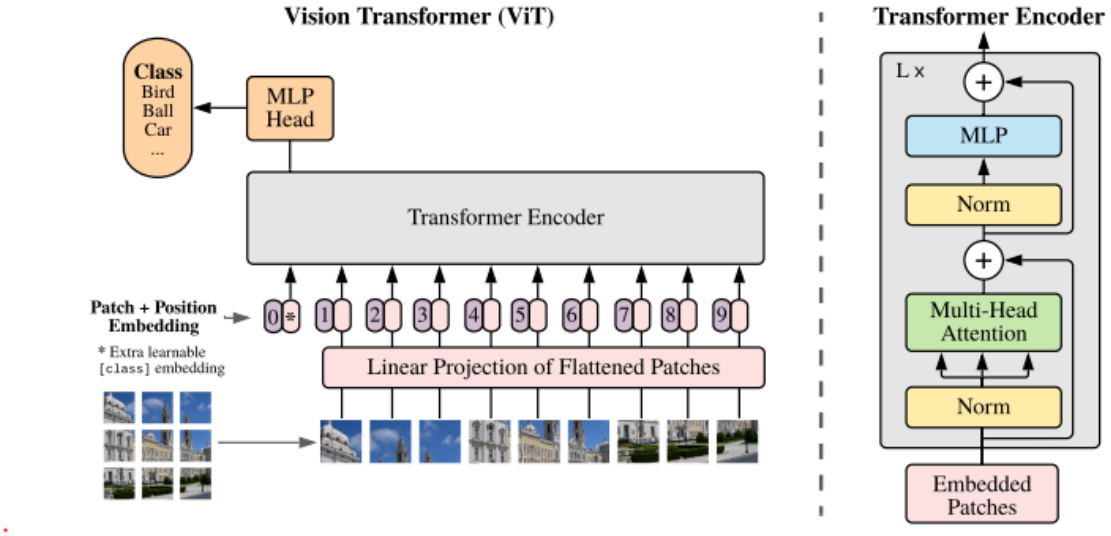

Fig 3.8 ViT Model overview [8]

So the main difference between the transformer encoder and VIT encoder is the type of input they process : the transformer encoder processes a sequence of tokens, while the VIT encoder processes a sequence of image patches. Additionally, the VIT encoder includes a patch embedding layer that maps the input image patches to a lower-dimensional space, which is not present in the original transformer encoder.

6.2. Contrastive learning

The idea of contrastive learning **[11]** is to have the ability to create a similarity metric that maps similar examples close together in the representation space and puts dissimilar examples far away.

In order to do contrastive pre-training we use 2 encoders to map each example in the pair to a representation in the high-dimensional space. The encoder is a vision transformer for image data and a transformer-based network for text data [3.4.2.4].

During the CLIP training process, after encoding both text and image , the two embeddings are then compared using a contrastive loss function, which encourages similar images and text to be mapped to nearby points in the embedding space .

Then a similarity metric is defined to compare the representations of the examples in the pair. This metric can be the Euclidean distance, cosine similarity, or other distance measures. In our case, we used cosine similarity because it only captures the orientation or direction of the vectors and the result will be in the range 1 , -1 which will maximaize the result between positive pairs and minimal between negative pairs . This property makes cosine similarity suitable for measuring similarity in high-dimensional spaces.

Finally, a contrastive loss function is used to optimize the encoder parameters with the goal to minimize the distance between positive pairs and maximize the distance between negative pairs (a positive example might be two different views of the same object, and a negative example is complete to different objects.

Fig 3.9 Contrastive Learning of Visual Representations [23]

6.3. Zero-shot classification

AI models always perform well with already seen images and classes that they were trained on, but what is the model's ability to generalize well to unseen classes during test time ?

and what if we have too many classes, and we cannot get enough right type and very clear data to train the model with ?

Fig 3.10 Example of training data

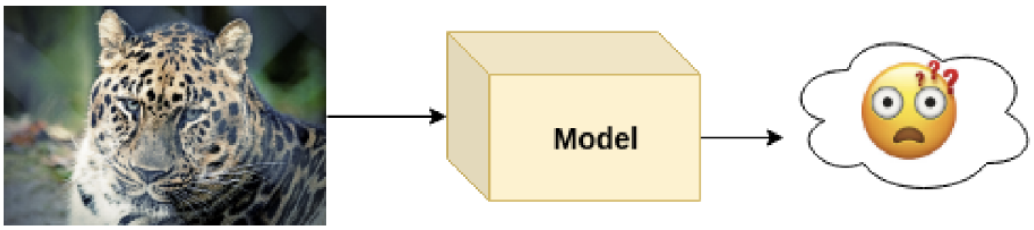

Fig 3.11 Example of test data of unseen classes

Most of the models, nowaday, are pretrained on N-shot learning where N is the number of data samples needed to train the model and the number of classes is less than 10 ,this type of learning is very useful when we are very limited on the number of training sample .
we have also another machine learning type which is many-shot learning where the model is trained on huge number of data for each class , and are tested on new examples from the same classes. Many-shot learning is useful when there is a large amount of data available for training, such as in image or speech recognition tasks.

Zero-shot learning is very useful in cases where we have many possible classes and it is not an easy situation for us to provide examples for all those classes .

With zero-shot learning we are trying to make the model focus and learn more discriminative visual features for better generalization (that is why we applied Contrastive Learning).

In zero-shot classification, the model is provided with a description or attributes of a new class of objects, rather than examples of actual objects from that class. The model then uses its pre-existing knowledge of related tasks to make predictions about the class of the object based on the provided description for example :

In image classification, a model may be trained to recognize a set of object classes such as dogs, cats, and birds. In zero-shot classification, the model can be tested on a new set of classes, such as horses or cows, without any additional training. Instead, the model is provided with a textual description or attributes of the new classes, such as "four-legged animal with hooves" for horses, and "four-legged animal with udders" for cows. The model can then use its pre-existing knowledge of related tasks to recognize and classify images of horses and cows based on their visual features and the provided textual description.

6.4. Conceptual Captions data-set

The CLIP model was pre-trained on a large-scale dataset called the "Conceptual Captions" dataset [4] that contains over three million image-caption pairs and each caption is a description of the image. The images in the dataset come from a wide range of sources: Wikipédia,web pages, books, and user-generated content.

"Conceptual Captions" dataset was chosen as the training dataset for CLIP because it provides a large and diverse set of visual concepts that are grounded in natural language descriptions. This allows the CLIP model to learn to associate words and phrases with visual features, and to reason about the relationships between different visual concepts.

7. Conclusion

In this chapter , we introduced two approaches based on multi-modal learning, where we discussed their implementation and the steps were clearly defined. In the second model we tried to cover all the points that caused a lack of performance of the application of multimodal learning via the first approach .

Chapter 3: Results & Discussions

1. Introduction

In order to perform contextual image classification we need to focus on all the actors of that image and the relationship between them, and to do that we have implemented two multimodal learning architectures that combine images and text to understand the context of an image.

Both models share the same general idea, but are implemented differently.

In this chapter we compare in depth both architectures and the limitations observed in the proposed approaches and address the potential reasons for the variations in performance of each model .

To compare between the two models we created a custom dataset that contains two columns, image_path and context . The first column was generated by selecting 100 random images from Flickr8k testing dataset [6], while the category column was filled manually.

2. Working environment

The choice of good tools and the way of using existing resources and methods are one of the most important parameters for the success of a project.

2.1. Agile methodology for AI project management

AI projects involve experimentation and refinement cycles , Agile practices helped us to break down the complex architecture into manageable tasks , and receive feedbacks from at the early stage of development .

For us we fixed the sprint period at 2 weeks and at the beginning of each sprint we do a sprint planning to select new and the tasks that needs to be completed with their priority .

At the end of each sprint we make a demo to demonstrate the completed work or we make a review if there is nothing technical to show .

At the end software engineering is an iterative process so we Repeat this process by selecting new tasks and starting a new Sprint. Continuously refine and prioritize the backlog based on feedbacks.

2.2 Tools and libraries

- **Google colab**

Due to the use of multimodal learning and the need to computational ressources we used google Colab Pro that provides access to computational resources, including CPU and GPU, which can be beneficial for running resource-intensive AI tasks and also Colab allowed us to collaborate on an effective way by having the ability to share code between us very easily .

- **Kaggle**

One of the significant advantages of Kaggle is its extensive library of datasets, which are contributed by the community and organizations. kaggle helps us save time and effort in data collection and cleaning.

In this sections , we describe the libraries that have been used in our project

- **TensorFlow**

Tensorflow is a free open source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks

- **NLTK**

Natural Language Toolkit is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language. It supports classification, tokenization, stemming, tagging, parsing, and semantic reasoning functionalities.

We used NLTK for preprocessing operations.

- **Keras**

is an open-source library that provides a python interface for artificial neural networks. Keras acts as an interface for the tensorflow library

- **Pandas**

Pandas is a software library written for the Python programming language for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables.

We used pandas to open , read and organize our datasets.

- **Pickle**

Pickle uses a powerful algorithm for serializing and de-serializing a python object structure
We used Pickle to save and load files .

- **Scikit-learn**

scikit-learn (or sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.
This library was used for classifiers and evaluation.

- **Transformer**

The transformer library is a state-of-the art machine learning library for pytorch , tensorflow and jax . It provides APIs and tools to easily download pretrained models . This helps to reduce computation costs, saves time and resources required to train a model from scratch.

3. Combined model results

To get back the combined model result, we proceeded by gathering Image captioning and Text classification model results to visualize the quality of each part.

3.1. Image captioning results

A good result is the one that provides a significant and comprehensive meaning and describes all the content of that image.
For image captioning there are two types of evaluation, automatic and human one .
For the automatic evaluation we used :

BLEU metric (Bilingual Evaluation Understudy) that measures the n-gram overlap between the generated caption and one or more reference captions, *the precision is calculated by dividing the number of matched unigrams, by the total number of unigrams in the generated sentence* .

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) measures the longest common subsequence (LCS) between the generated summary and the reference summary ,the score would be the length of the LCS divided by the length of the reference sentence .

Here is a comparison between the different scores obtained after the generation of captions on our test set

Blue-1	Blue-2	Blue-3	Blue-4	Rouge-L
0.5820	0.4418	0.3473	0.2790	0.2661

Tab 1 Image captioning model results

We noticed that the scores started well then they decreased on the BLUE-N evaluation and this is normal since we are comparing consecutive similar words. The best way to compare image captioning is still human evaluation .

Here is an example how our model did to generate a caption for this image :

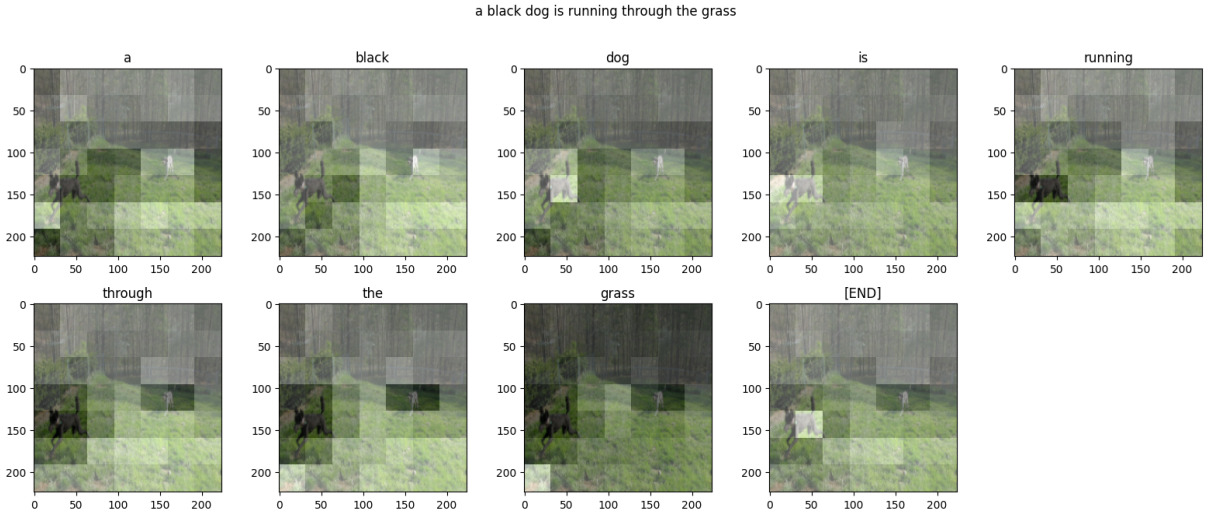

Fig 5.1 Image from our test set

Reference Sentence: "A black and white dog is running through the grass ."
Generated Sentence: "A black dog is running through the grass"

Blue-1	Blue-2	Blue-3	Blue-4	Rouge-L
1.0	0.857	0.833	0.2	0.889

Tab 2 Calculating bleu and rouge for Fig 5.1

In Tab 2 we are calculating Bleu and Rouge scores using the reference and the generated caption for the previous example [fig 5.1].

3.2. Text classification results

To achieve the best performance for the text classifier model, we experimented with three algorithms : logistic regression (LR), support vector machine (SVM) and Naive bayes (NB), we tested each algorithm with two types of vectorizer : CountVectorizer and TFIDF.
The summarized evaluation results can be found in the table below.

Algorithms	Vectorizer	Accuracy	Precision	Recall	F1 Score
LR	Count Vectorizer	0.8926	0.8910	0.8926	0.8912
LR	TFIDF Vectorizer	0.9017	0.9003	0.9017	0.9003
SVM	Count Vectorizer	0.8792	0.8775	0.8792	0.8781
SVM	TFIDF Vectorizer	0.8907	0.8895	0.8907	0.8869
NB	Count Vectorizer	0.8906	0.8882	0.8906	0.8877
NB	TFIDF Vectorizer	0.8426	0.8522	0.8426	0.8162

Tab 3 Text classification model results

In preprocessing both lemmatization and stemming, gave us the same results.
We ended up choosing the logistic regression with TFIDF vectorizer due to its scalability and fast performance compared to SVM which tends to scale poorly and is considered to be more resource and computation greedy.

3.3. Evaluation

The model showed a good performance when it comes to images close to the images in our training set but when we pass an image far from the images from the training the caption can be out of context , the biggest disadvantage of this model is that the quality of the caption generated can highly affect the final results , so if the captions are inaccurate or fail to capture the relevant context, it may introduce noise or misleading information and it will lead directly

to a wrong result whatever is the accuracy of the text classification model .The captions are limited in size (depending on the captions used in training) so sometimes a loss of information will happen if the image is too complex and full of information .

and when it comes to computational resources this method was very time and resources consuming.

The figures below , shows how this model performed on some of the testing set images.

In Fig 5.2, the combined model correctly predicts the given image as ‘Sport’, as we loaded the image , generate a caption for it, then classify it with logistic regression

Fig 5.2 Combined Model prediction 1

In the second image [fig 5.3] the predicted_category ‘politics’ was a wrong prediction as the true category was ‘sport’.

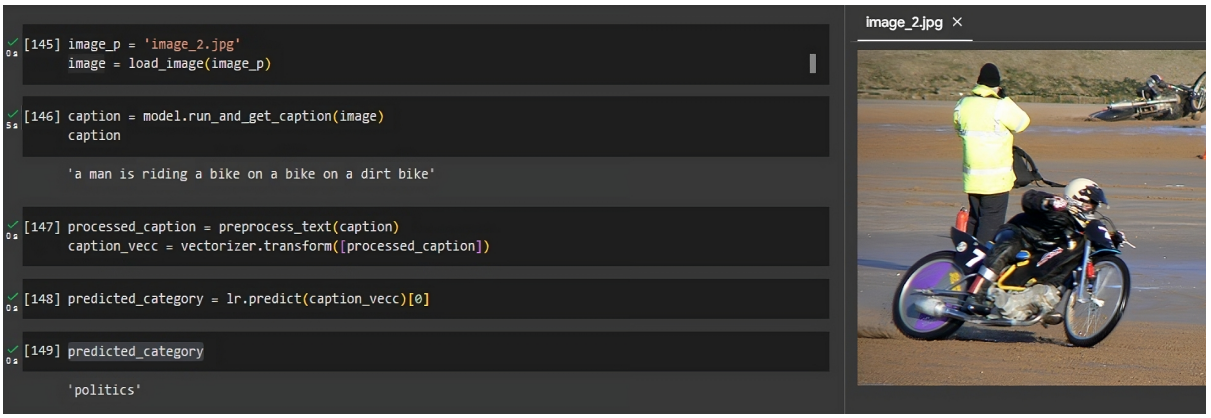

Fig 5.3 Combined Model prediction 2

In the next image Fig 5.4 we tried to use an unseen image different from the testing set images that we used previously.

the model predicted that image as ‘lifestyle’ , we can see that the generated caption is not good enough, as in this specific image corresponds more to ‘entertainment’ or ‘sport’

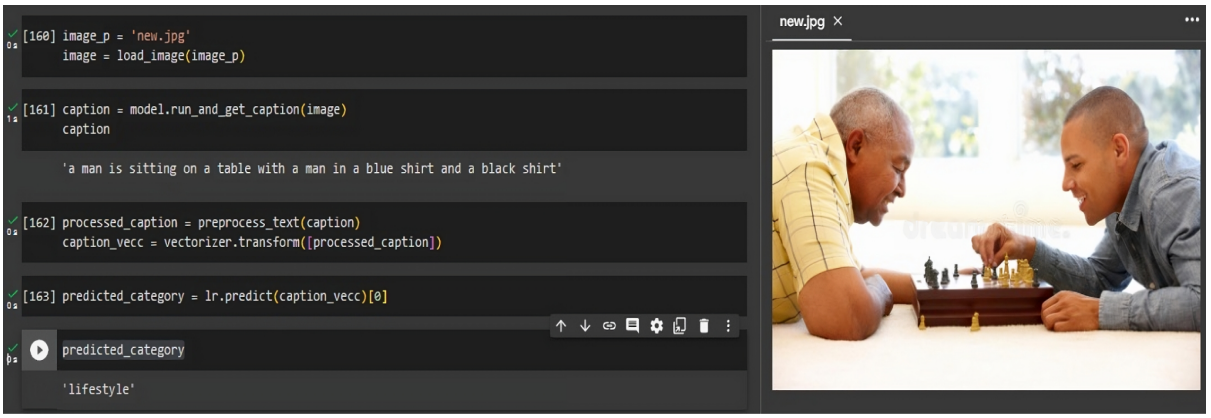

Fig 5.4 Combined Model prediction test on unseen_image

4. CLIP neural network results

CLIP mainly was invented to search images based on text input but we still can use transfer learning to make it do contextual image classification for our classes. We can transfer them to sentences .

To do this,we simply concatenate the class name to the end of a fixed sentence template like "photos related to", like so:

education -> photos related to education

health care -> photos related to health care

sport -> photos related to sport

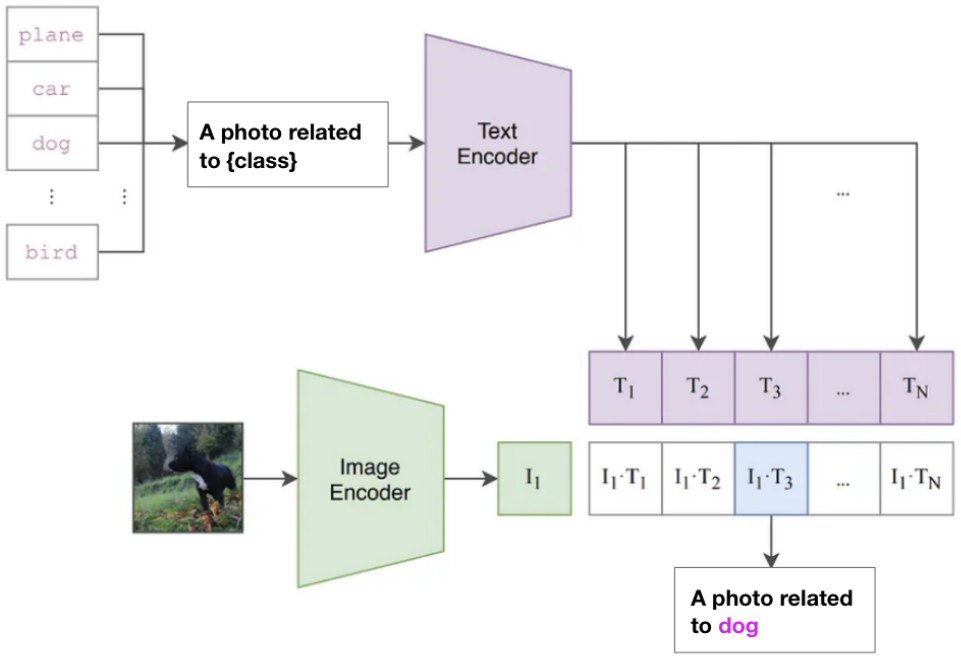

Fig 5.5 Using clip for image classification

When it comes to CLIP , the results were better in the custom dataset.

Using CLIP with contextual incorporation gave us 80% accuracy on the same dataset used in the previous model. Fig 5.6 shows that accuracy.

```
[9]  # Calculate accuracy
     accuracy = (correct_predictions / total_images) * 100
     print(f"Accuracy: {accuracy:.2f}%")

     Accuracy: 80.00%
```

Fig 5.6 CLIP accuracy on the custom dataset

We believe that the accuracy is affected by the quality of captions provided by Flickr8k dataset and their respective categories (referring to the custom dataset).

4.1 Evaluation

After passing our test set to the CLIP neural network model we noticed that the model had a better performance when we tried it on an image that doesn't belong to our training set and this is due to zero-shot learning that exists in CLIP. And also, the use of the vision transformer directly captured both local and global relationships in images through self-attention, allowing them to effectively model long term dependencies more than CNN which is more efficient when it comes to one main visual feature [9].

We tested CLIP with the custom dataset of 100 samples. Some of the results are in the figures below.

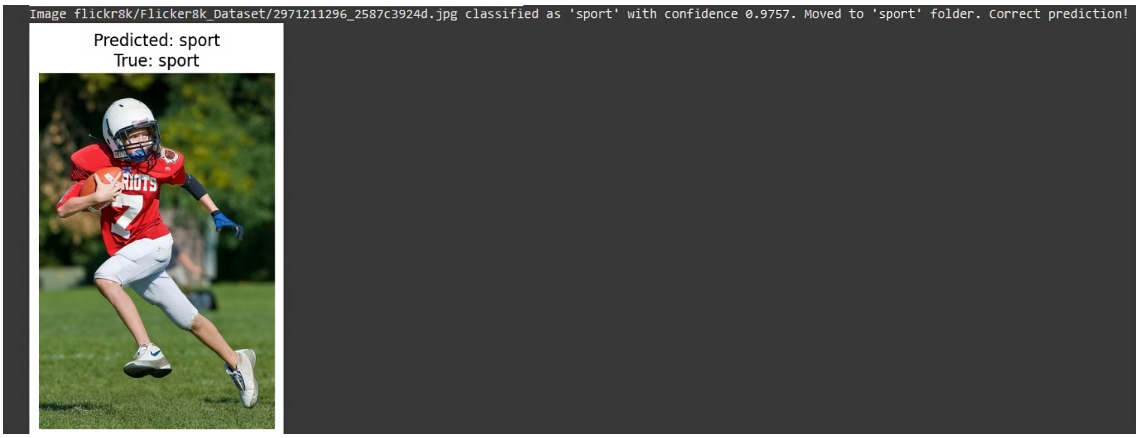

Fig 5.7 CLIP image nb1 prediction on testing set

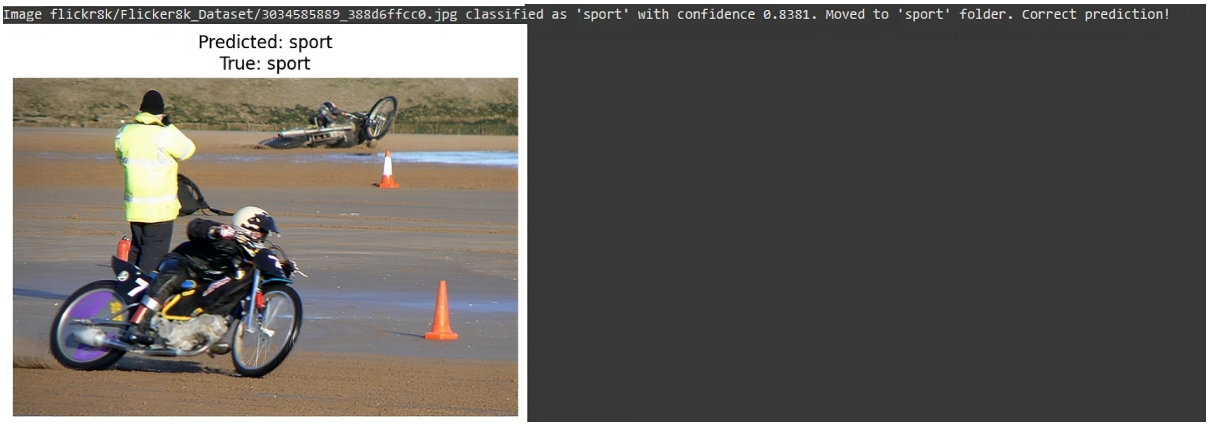

Fig 5.8 CLIP image nb2 prediction on testing set

Fig 5.9 shows how well clip can do on unseen images , chess can be considered as a sport game.

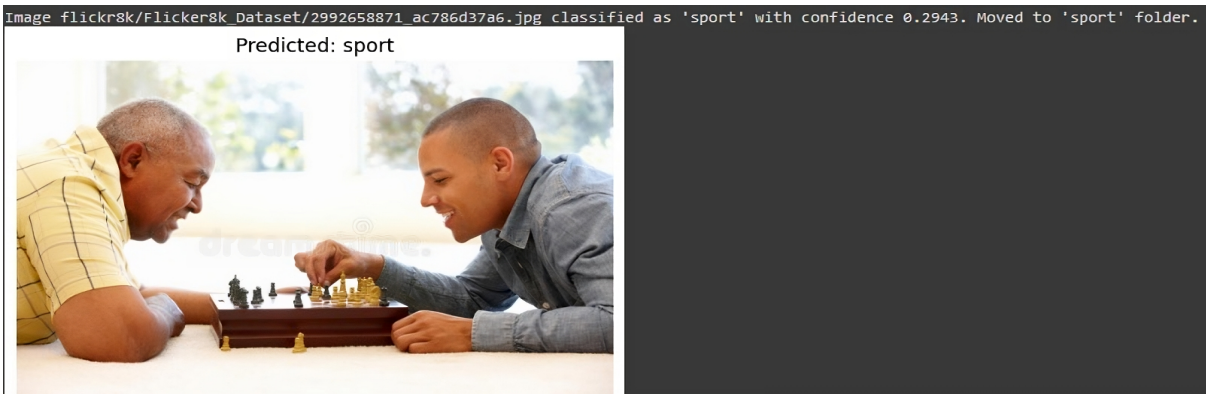

Fig 5.9 CLIP prediction on an unseen image

5. Final Result

As demanded from IcosNet, we created a web application that takes a set of images uploaded by the user and classify them instantly based of the context of each image

For the backend of the application, we used django framework and react js library for the frontend.

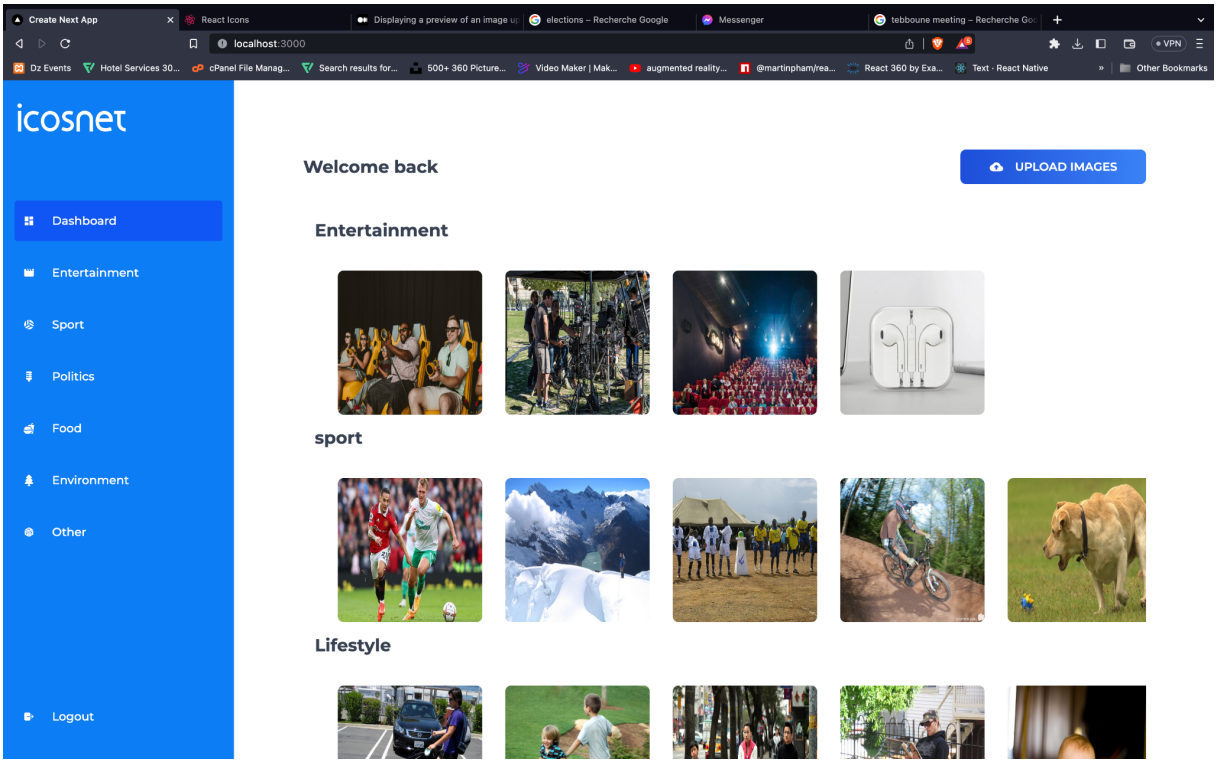

Fig 5.10 Classified images based on context after upload 01

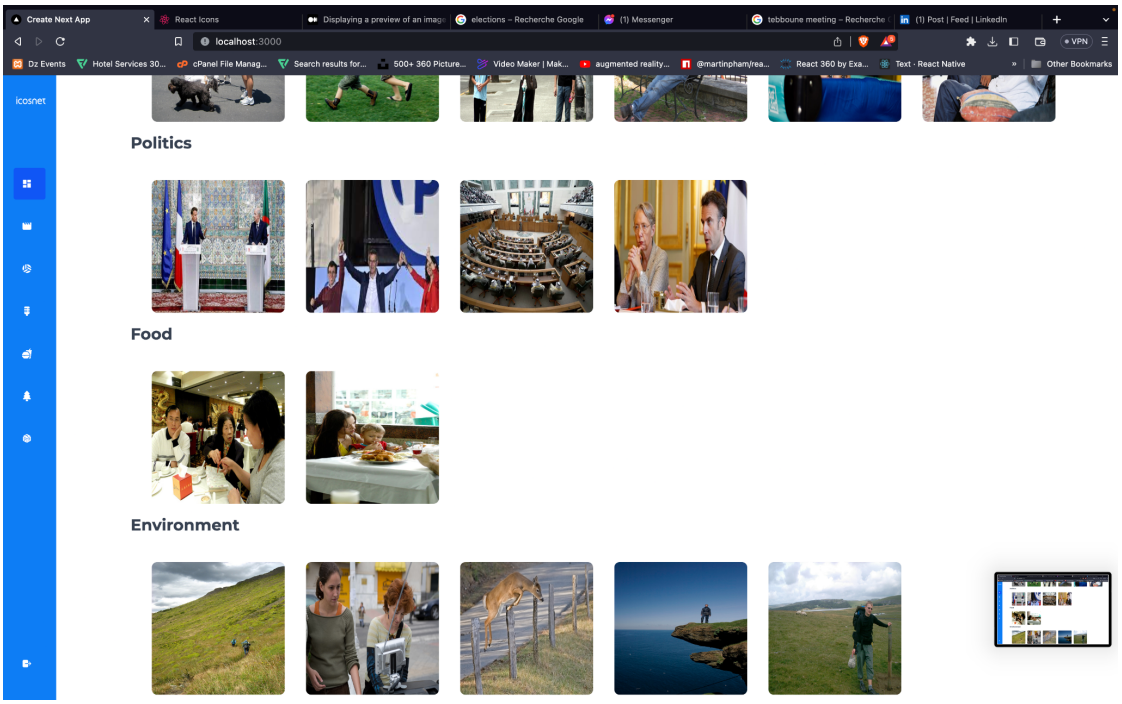

Fig 5.11 Classified images based on context after upload 02

6. Conclusion

In this chapter, we discussed the results that we got after training both models and the main reasons that affected the final result of each , and how the model behaves to the different complex images that we gave .

By the end we presented a web application that has the desired behavior of Icosnet Ibox , that takes a set of images as an input and the output is a set of images classified based on context .

General Conclusion

The goal of this work was to give icosnet IBox client the ability to classify images in their storage based on the context of the image , and in order to classify images by context and after some research we found that the right way to do it is to focus on all the parts of the image and the relationship between them .

until now many researchers have explored the use of attention mechanisms in CNNs to capture contextual information and with the apparition of multimodal learning that achieved the state of the art performance on multiple benchmark datasets , we aimed to use a combination of textual captions and images during the training in order to simulate the human brain behavior to know the context of the image .

This work was preceded by an introduction of the current state of the art and some background knowledge that we should have before getting into multimodal learning and training models on both texts and images .

during this work we used 2 ways in order to do the contextual classification , the first one was a combination of image captioning model that has been built with CNNs and transformer based model after this we added on the top of it a text classification model using logistic regression to extract the context from the caption .

the seconde way was with use CLIP neural network that achieved competitive zero-shot performance on a great variety of image classification datasets , and this what gived us the ability to classify images from unseen classes correctly .

and due to limitation of resources , time and the correct datasets we couldn’t get a more accurate result if we got better training data, especially textual description quality .

At the end we realized a web application that simulated the same behavior that icosnet wanted for this client , and our solution will be implemented in the existing ibox services.

This project was very helpful for us because it gives us the ability to explore the potentials of multimodal learning in image classification and also we explored the way to train models on different modalities (text,images).

for further improvement in the future we would like to get more data with better quality of captioning this will help to get a better results in the future

and big satisfaction was to see our solution get implemented on a real world product to give the users a better experience which is the goal of this project.

BIBLIOGRAPHY

[1] A. Vaswani *et al.*, "Attention Is All You Need." arXiv, Dec. 05, 2017. doi: 10.48550/arXiv.1706.03762.

[2] Plested, Jo & Gedeon, Tom, "An Analysis of the Interaction Between Transfer Learning Protocols in Deep Neural Networks" https://www.researchgate.net/publication/337922940_An_Analysis_of_the_Interaction_Between_Transfer_Learning_Protocols_in_Deep_Neural_Networks

[3] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based Image Description Evaluation." arXiv, Jun. 02, 2015. doi: 10.48550/arXiv.1411.5726.

[4] S. Changpinyo, P. Sharma, N. Ding, and R. Soricut, "Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts." arXiv, Mar. 30, 2021. doi: 10.48550/arXiv.2102.08981.

[5] J. Plested and T. Gedeon, "Deep transfer learning for image classification: a survey." arXiv, May 19, 2022. doi: 10.48550/arXiv.2205.09904.

[6] "Flickr 8k Dataset." https://www.kaggle.com/datasets/adityajn105/flickr8k (accessed Jun. 27, 2023).

[7] A. A. Elngar *et al.*, "Image Classification Based On CNN: A Survey," *Journal of Cybersecurity and Information Management*, vol. Volume 6, no. Issue 1, p. PP. 18-50, Oct. 2021, doi: 10.54216/JCIM.060102.

[8] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *Journal of Artificial Intelligence Research*, vol. 47, pp. 853–899, Aug. 2013, doi: 10.1613/jair.3994.

[9] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." arXiv, Jun. 03, 2021. doi: 10.48550/arXiv.2010.11929.

[10] H. Sak, A. Senior, and F. Beaufays, "Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition." arXiv, Feb. 05, 2014. doi: 10.48550/arXiv.1402.1128.

[11] A. Radford *et al.*, "Learning Transferable Visual Models From Natural Language Supervision." arXiv, Feb. 26, 2021. doi: 10.48550/arXiv.2103.00020.

[12] *MIT 6.S191: Recurrent Neural Networks, Transformers, and Attention*, (Mar. 17, 2023). Available: https://www.youtube.com/watch?v=ySEx_Bqxvvo

[13] T.-Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context." arXiv, Feb. 20, 2015. doi: 10.48550/arXiv.1405.0312.

[14] R. Misra, "News Category Dataset." arXiv, Oct. 06, 2022. doi: 10.48550/arXiv.2209.11429.

[15] A. Ajit, K. Acharya, and A. Samanta, "A Review of Convolutional Neural Networks," in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, Feb. 2020, pp. 1–5. doi: 10.1109/ic-ETITE47903.2020.049.

[16] A. Shrestha and A. Mahmood, "Review of Deep Learning Algorithms and Architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019, doi: 10.1109/ACCESS.2019.2912200.

[17] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.

[18] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks." arXiv, Dec. 14, 2014. doi: 10.48550/arXiv.1409.3215.

[19] K. Xu *et al.*, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention." arXiv, Apr. 19, 2016. doi: 10.48550/arXiv.1502.03044.

[20] M. Stefanini, M. Cornia, L. Baraldi, S. Cascianelli, G. Fiameni, and R. Cucchiara, "From Show to Tell: A Survey on Deep Learning-Based Image Captioning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 539–559, Jan. 2023, doi: 10.1109/TPAMI.2022.3148210.

[21] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "SPICE: Semantic Propositional Image Caption Evaluation." arXiv, Jul. 29, 2016. doi: 10.48550/arXiv.1607.08822.

[22] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in Vision: A Survey," *ACM Comput. Surv.*, vol. 54, no. 10s, pp. 1–41, Jan. 2022, doi: 10.1145/3505244.

[23] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, "With a Little Help from My Friends: Nearest-Neighbor Contrastive Learning of Visual Representations," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 9568–9577. doi: 10.1109/ICCV48922.2021.00945.

[24] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks," *Artif Intell Rev*, vol. 53, no. 8, pp. 5455–5516, Dec. 2020, doi: 10.1007/s10462-020-09825-6.

[25] "What is Convolutional Neural Network — CNN (Deep Learning) | by Nafiz Shahriar | Medium." https://nafizshahriar.medium.com/what-is-convolutional-neural-network-cnn-deep-learning-b3921bdd82d5 (accessed Jun. 27, 2023).

[26] P. Karkare, "Understanding Recurrent Neural Networks in 6 Minutes," *AI Graduate*, Sep. 13, 2019. https://medium.com/x8-the-ai-community/understanding-recurrent-neural-networks-in-6-minutes-967ab51b94fe (accessed Jun. 27, 2023).