CS420 Project:

VIETNAMESE IMAGE CAPTIONING

Lecturer: PhD.Mai Tiến Dũng & Master Đỗ Văn Tiến

Our Group:

21520531 – Nguyễn Hà Anh Vũ
21520648 – Nguyễn Đinh Minh Chí
21521993 – Hứa Bảo Duy

TABLE OF CONTENTS

I

Introduction

II

Related works

III

Methods

IV

Experiments & Results

V

Conclusion & Future works

VI

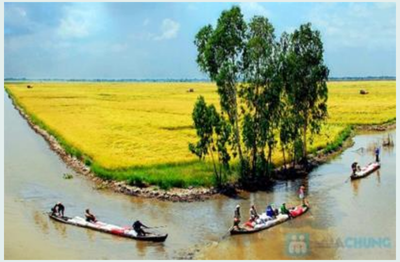
Demo

I

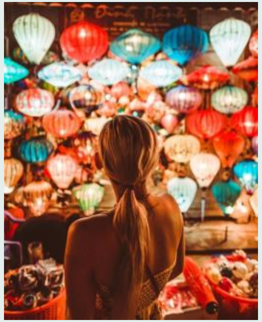
INTRODUCTION

Definition

IMAGE CAPTIONING is the task of describing the visual content of the image in natural language, employing a visual understanding system and a language model capable of generating meaningful and syntactically correct sentences.



ba chiếc thuyền đang di chuyển ở trên con sông



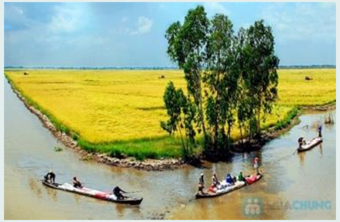
người phụ nữ đang nhìn vào những chiếc đèn lồng

Problem identification

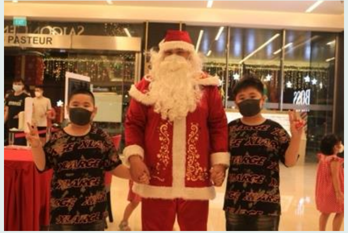
Input	+ A dataset consisting of images, each paired with one or more captions that describe its content. Each image may contain one or more objects, contexts, actions, or other elements that the model needs to identify and understand. + A new image doesn't have caption.	+ $D = \{(I_i, C_i)\}_{i=1}^{N}$ with $\begin{cases} I_i \in F = R^d \\ C_i = \{c_{i1}, c_{i2}, \dots c_{im}\} \end{cases}$ + $I \in F$
Output	A textual caption describing the visual content of the image, including information on objects, actions, context, and other relevant details, aiming to accurately and naturally convey what is happening in the scene.	$\hat{c} = f(I)$

Example

Input

Image	Caption
	1. ba chiếc thuyền đang di chuyển ở trên con sông 2. có ba con thuyền đang di chuyển trên con sông 3. trên dòng sộng có ba con thuyền đang di chuyển 4. ba con thuyền đang di chuyển bên một cánh đồng lúa 5. ba chiếc thuyền đang chuyển động trên một con sông
…	…

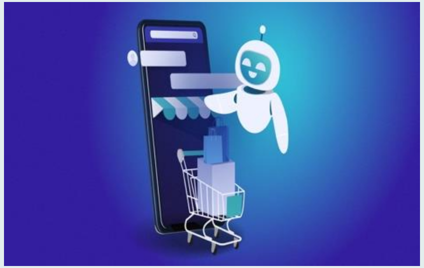
New image



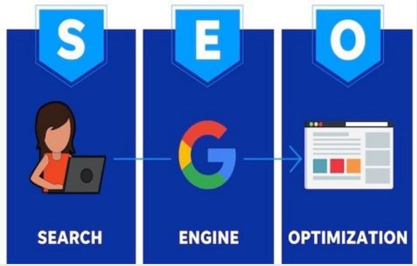
Output: " ở hai bên của ông già noel là hai cậu bé "

Application

Support people with visual impairments



Describe product images in the e-commerce field



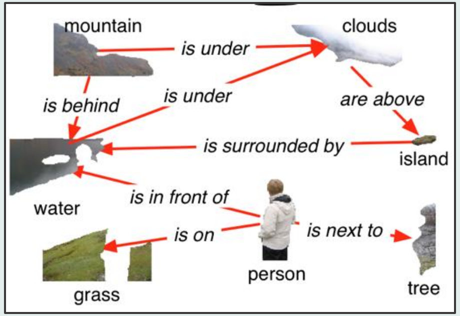
Optimize search quality for image-based search engines

Challenges

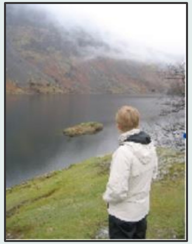
Complex scene understanding: Accurately interpreting images with multiple objects, intricate interactions, and diverse contexts is difficult, need to not only describe salient elements but also subtle details.

Main object

Relationship with other objects



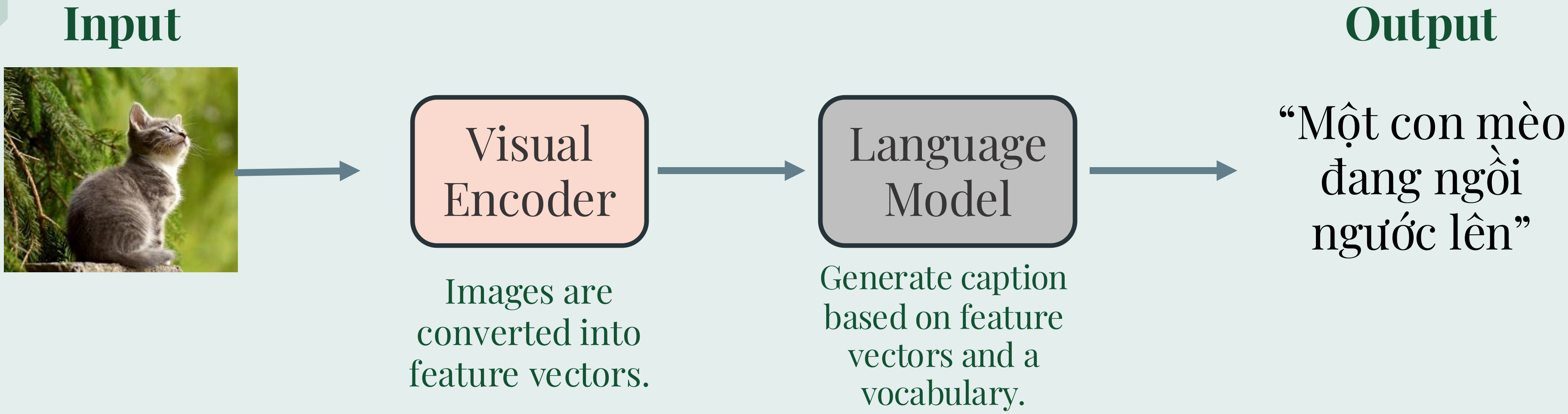
Language Fluency and Diversity: generated captions are grammatically correct, coherent, and diverse is challenging.



Caption 1: Người đàn ông mặc áo khoác đang nhìn xuống hồ.
Caption 2: Một người áo trắng, tóc vàng đứng trước mặt hồ.

II
Related works

Pipeline



Input

Visual Encoder

Images are converted into feature vectors.

Language Model

Generate caption based on feature vectors and a vocabulary.

Output

“Một con mèo đang ngồi ngước lên”

In the Visual Encoding step, images are encoded into one or more feature vectors, which serve as input for the second generative step, called the Language Decoding. This step generates a sequence of words that are decoded based on a given vocabulary.

Pipeline

Input

Visual Encoding

1. Non-Attentive
(Global CNN Features)
2. Additive Attention:
- Grid-based
- Region-based

3. Graph-based Attention
4. Self-Attention:
- Region-based
- Patch-based
- Image-Text Early Fusion

Language Models

1. LSTM-based:
- Single-layer
- Two-layer

2. CNN-based
3. Transformer-based
4. Image-Text Early Fusion
(BERT-like)

Output

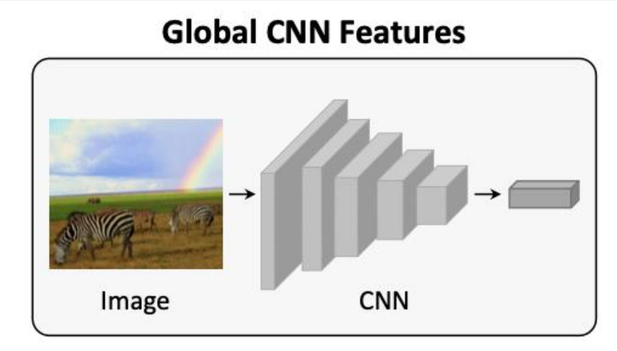
"Một con mèo đang ngồi ngước lên"

Training Strategies

1. Cross-Entropy Loss
2. Masked Language Model
3. Reinforcement Learning
4. Vision Language Pretraining

Visual Encoding

Non-Attentive - Global CNN Features



the activation of one of the last layers of a CNN is employed to extract high-level representations, which are then used as a conditioning element for the language model

This is the approach employed in the seminal "**Show and Tell**" paper

Output of GoogleNet [24] is fed to the initial hidden state of the language model.

+ Karpathy et al used global features extracted from AlexNet as the input for a language model.
+ Mao et al and Donahue et al. injected global features extracted from the VGG network at each time-step of the language model.
+ Rennie et al. [38] introduced the FC model, in which images are encoded using a ResNet-101, preserving their original dimensions.

STRENGTH & DRAWBACKS

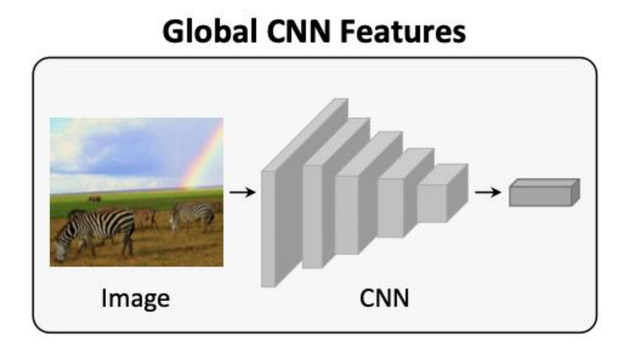
Non-Attentive - Global CNN Features

STRENGTH:

- Simplicity, Low computational resource consumption and easy to develop.
- Compactness of representation

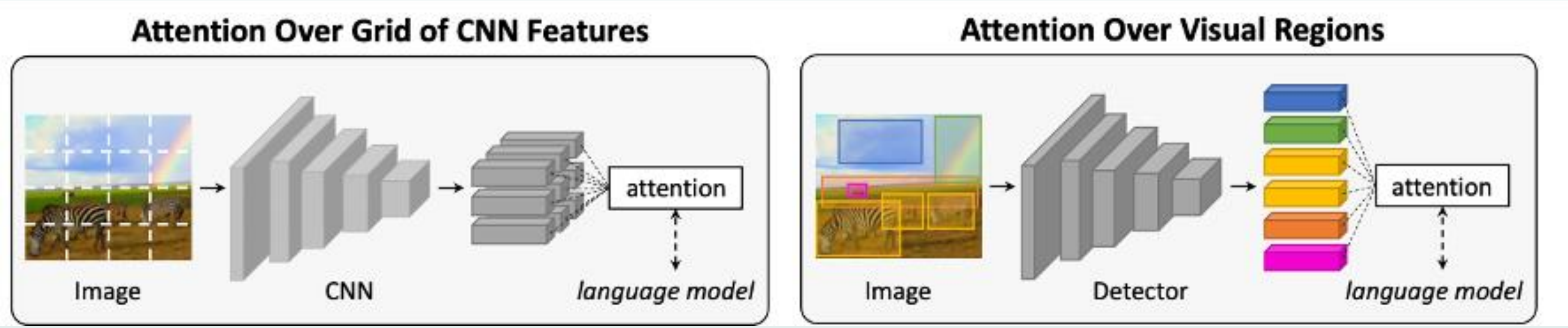
DRAWBACKS:

- Excessive compression of information
- Lacks granularity



Visual Encoding

Additive Attention



Attention Over Grid of CNN Features

Attention Over Visual Regions

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention (2016)

Kelvin Xu
Jimmy Lei Ba
Ryan Kiros
Kyunghyun Cho
Aaron Courville
Ruslan Salakhutdinov
Richard S. Zemel
Yoshua Bengio

STRENGTH & DRAWBACKS

Additive Attention

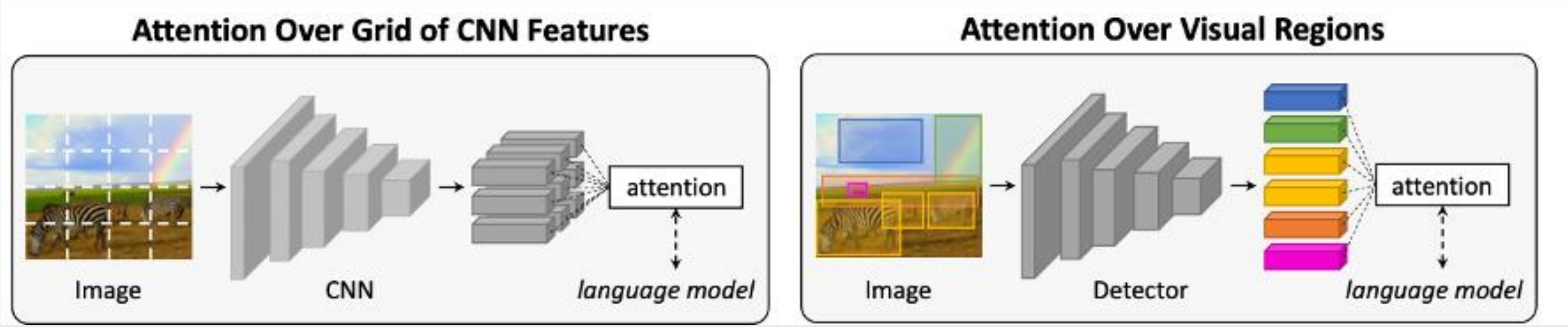
STRENGTH:

+ focus on important areas of the image in each step of generating the description, thereby creating a more detailed and accurate description.

+ Improves the ability to recognize small objects

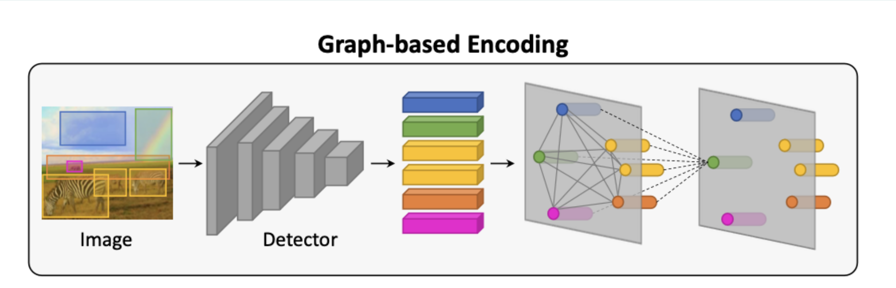
DRAWBACKS:

+ Consumes computational resources to calculate numbers for many areas.



Visual Encoding

Graph-based Encoding



“Exploring Visual Relationship for Image Captioning,” in ECCV, 2018
T. Yao, Y. Pan, Y. Li, and T. Mei

STRENGTH & DRAWBACKS

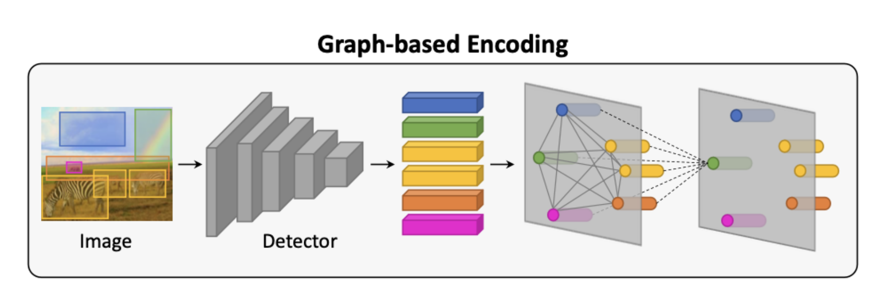
Graph-based Encoding

STRENGTH:

- Information exchange
- Semantic information

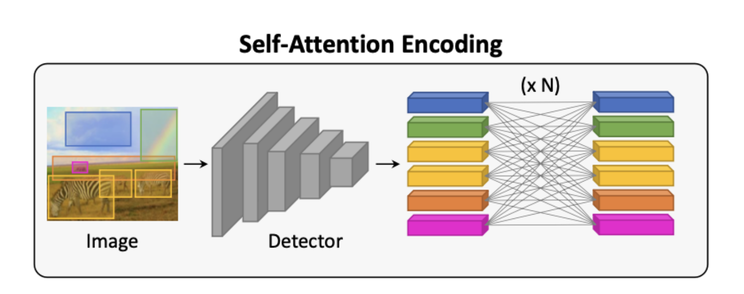
DRAWBACKS:

- Interactions between visual features



Visual Encoding

Self-Attentive Methods



“Learning to Collocate Neural Modules for Image Captioning,” in ICCV, 2019.
X. Yang, H. Zhang, and J. Cai,

STRENGTH & DRAWBACKS

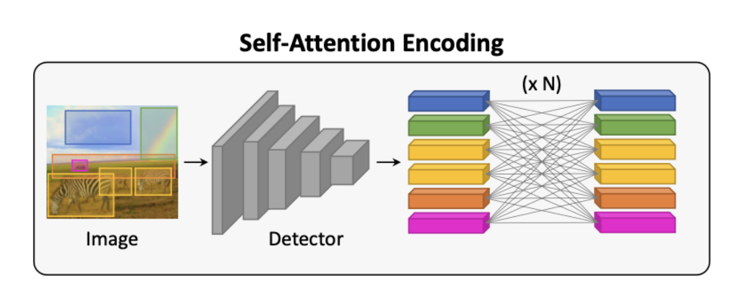
Self-Attentive Methods

STRENGTH:

- Automatically learns connections between different parts of an image without depending on specific locations.
- Efficiency and performance allow modeling of complex interacting complexes without the need for schematic or network structures.

DRAWBACKS:

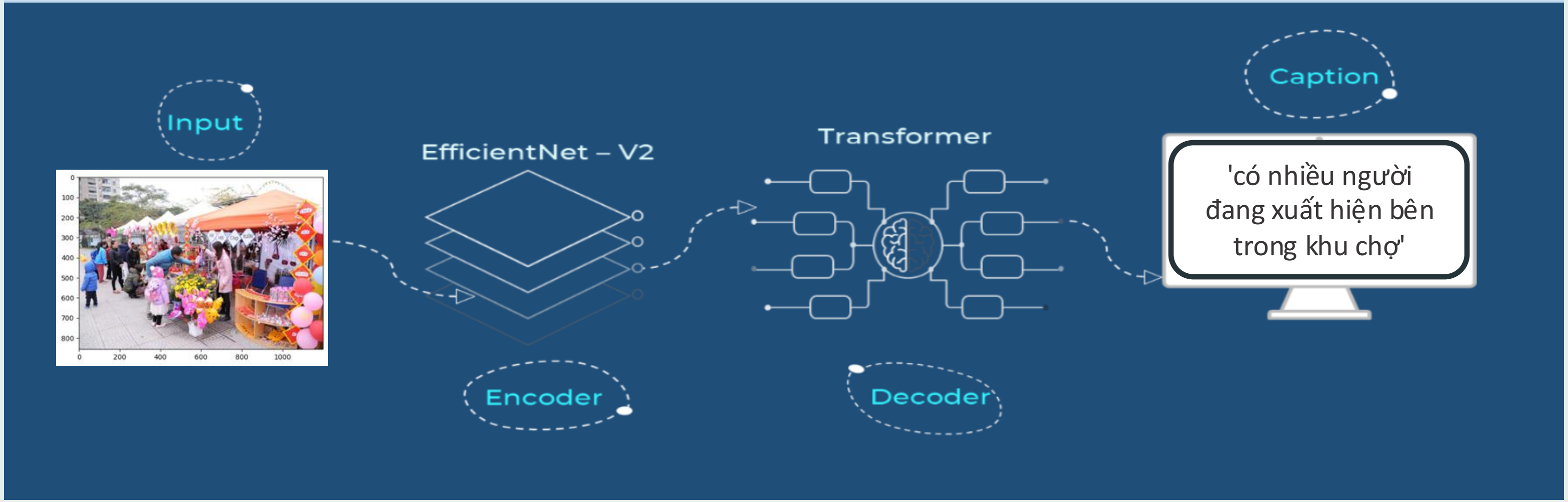
- More difficult to train, requiring more memory and computation than traditional attention models.



III

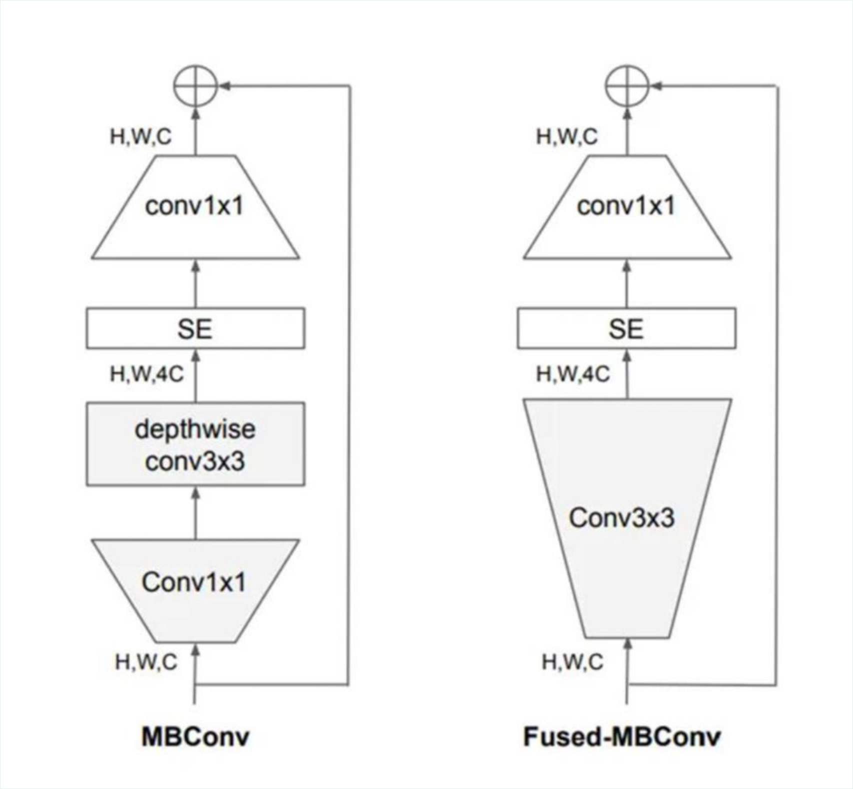
METHODS

1. EfficientNet_v2 + Transformer

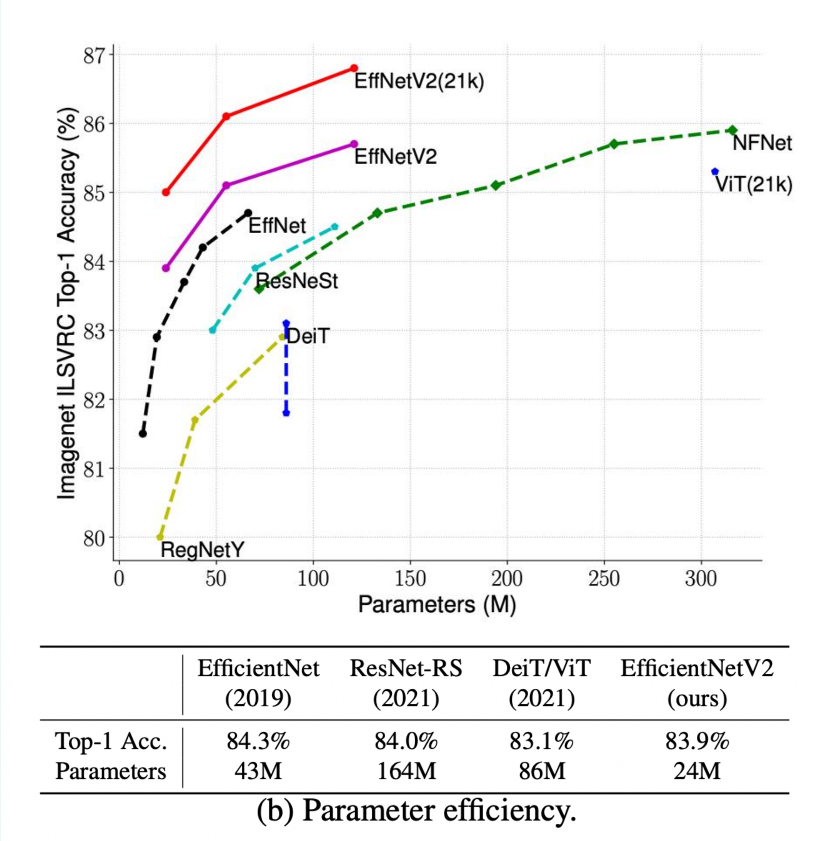


'có nhiều người đang xuất hiện bên trong khu chợ'

1. EfficientNet_v2 + Transformer



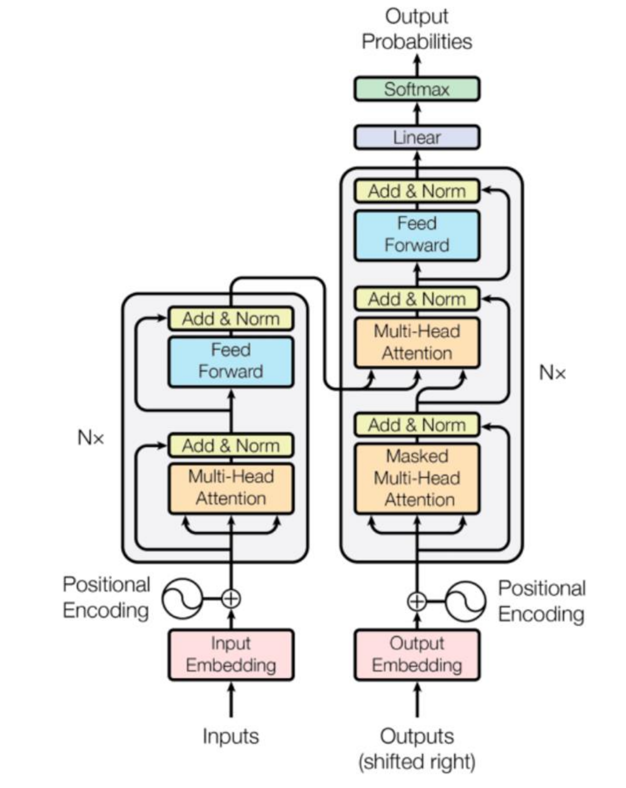
1. EfficientNet_v2 + Transformer



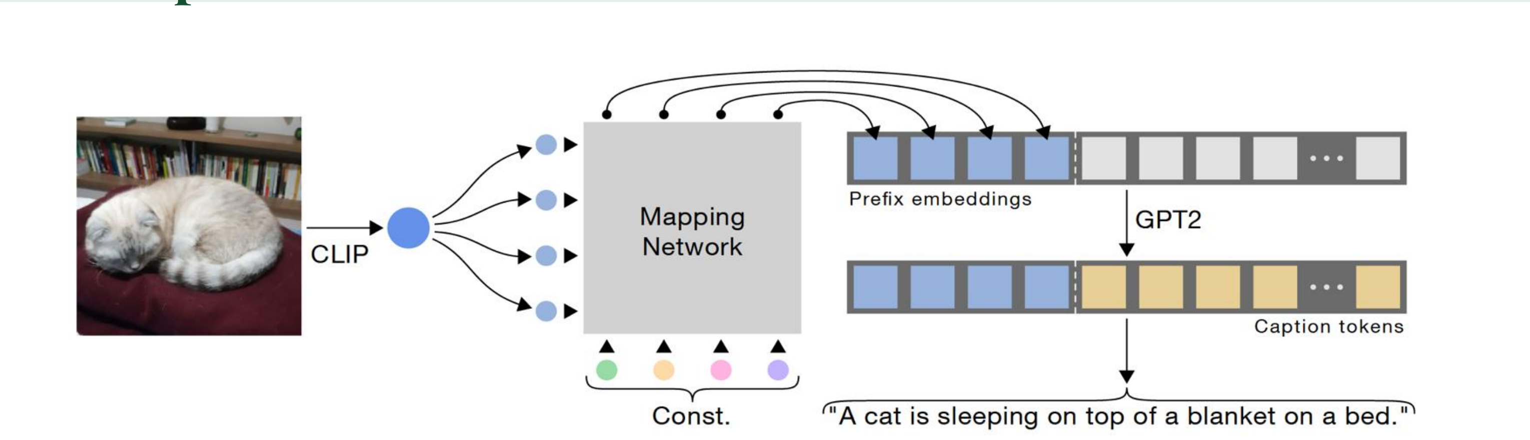
	EfficientNet (2019)	ResNet-RS (2021)	DeiT/ViT (2021)	EfficientNetV2 (ours)
Top-1 Acc.	84.3%	84.0%	83.1%	83.9%
Parameters	43M	164M	86M	24M

(b) Parameter efficiency.

1. EfficientNet_v2 + Transformer



2. CLIPCap

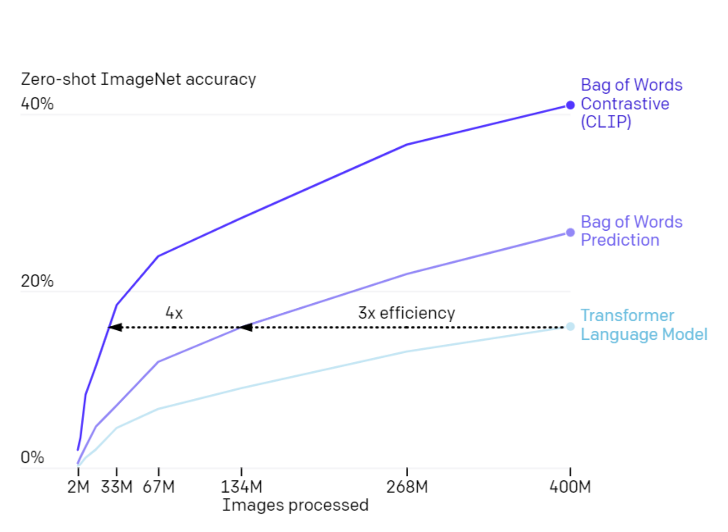


ClipCap model has 3 main components: CLIP, Mapping Network and GPT-2
- Pretrain CLIP (Contrastive Language–Image Pre-training) model to extract semantic information of the image.
- Pretrain GPT-2 to generate caption from that semantic information.
- Mapping Network (key component of paper/model) to transform encoding (output of CLIP) into word embedding (input of GPT-2).

The training process of this model is the Mapping Network training process, because CLIP and GPT-2 are pretrained models with very large sizes that do not need to be retrained but still achieve high results.

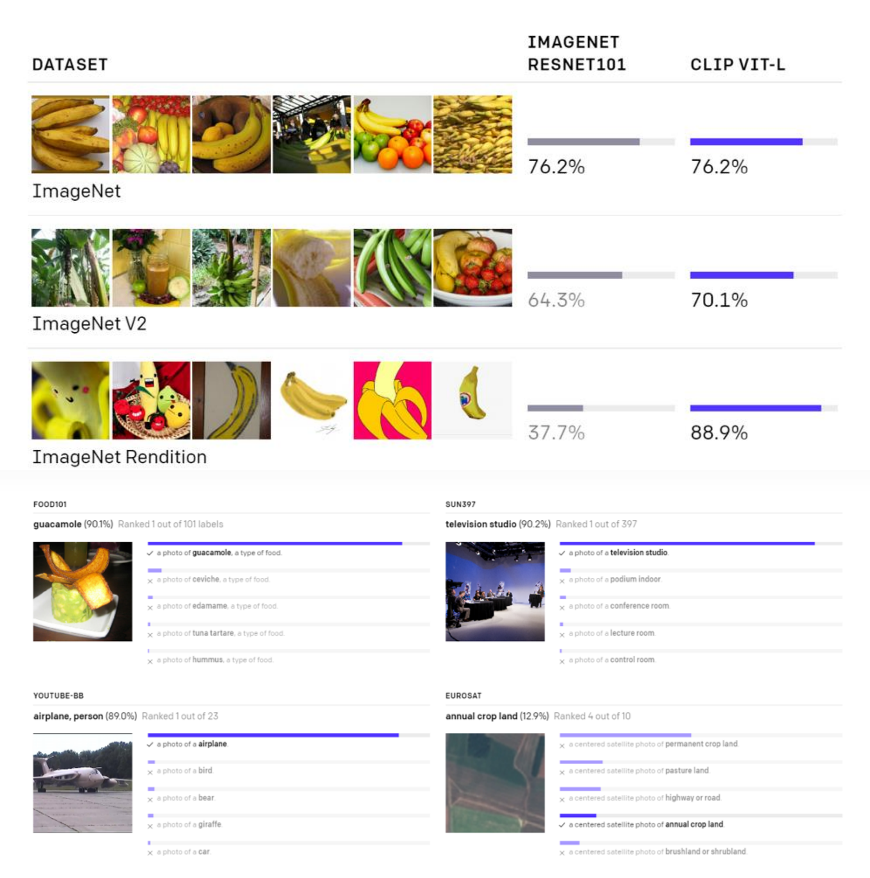
2. CLIPCap

CLIP



CLIP is a powerful multimodal Vision-Language deep learning model by OpenAI in 2021, trained on 400,000,000 (image, text) pairs.

The basic idea of CLIP is to encode image and text pairs in the same representation, so that these concepts can be compared,



DATASET	IMAGENET RESNET101	CLIP VIT-L
ImageNet	76.2%	76.2%
ImageNet V2	64.3%	70.1%
ImageNet Rendition	37.7%	88.9%

FOOD101

guacamole (90.1%) Ranked 1 out of 101 labels

✓ a photo of **guacamole**, a type of food.

× a photo of **ceviche**, a type of food.

× a photo of **edamame**, a type of food.

× a photo of **tuna tartare**, a type of food.

× a photo of **hummus**, a type of food.

SUN397

television studio (90.2%) Ranked 1 out of 397

✓ a photo of a **television studio**.

× a photo of a **podium indoor**.

× a photo of a **conference room**.

× a photo of a **lecture room**.

× a photo of a **control room**.

YOUTUBE-BB

airplane, person (89.0%) Ranked 1 out of 23

✓ a photo of a **airplane**.

× a photo of a **bird**.

× a photo of a **bear**.

× a photo of a **giraffe**.

× a photo of a **car**.

EUROSAT

annual crop land (12.9%) Ranked 4 out of 10

× a centered satellite photo of **permanent crop land**.

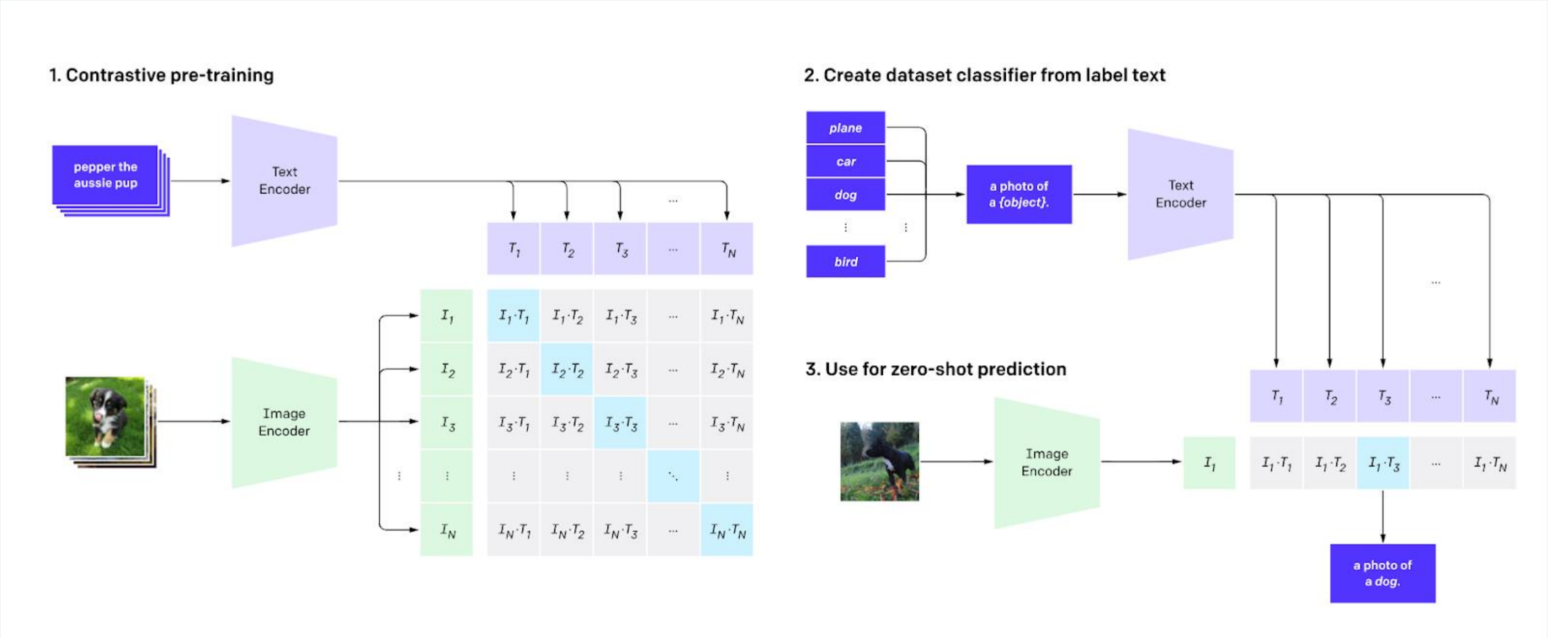
× a centered satellite photo of **pasture land**.

× a centered satellite photo of **highway or road**.

✓ a centered satellite photo of **annual crop land**.

× a centered satellite photo of **brushland or shrubland**.

2. CLIPCap

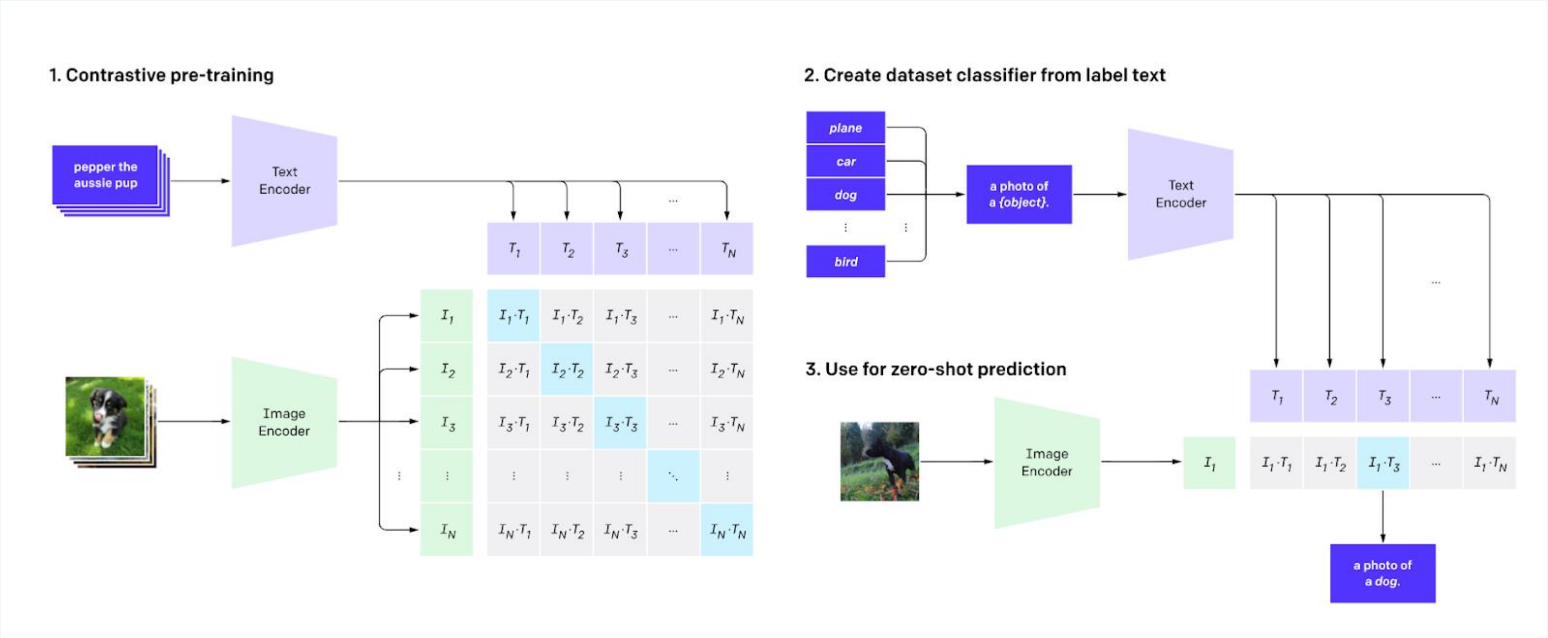

1. Contrastive pre-training
pepper the aussie pup
Text Encoder
Image Encoder
2. Create dataset classifier from label text
plane
car
dog
bird
a photo of a {object}.
Text Encoder
3. Use for zero-shot prediction
Image Encoder
I_1
T_1 T_2 T_3 ... T_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
a photo of a dog.

	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
⋮	⋮	⋮	⋮	⋱	⋮
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

Idea: Encode image and text in the same representation so they can be compared

- ➔ Optimize the connection between image and text
- ➔ **CLIP is a bridge between computer vision and natural language processing.**

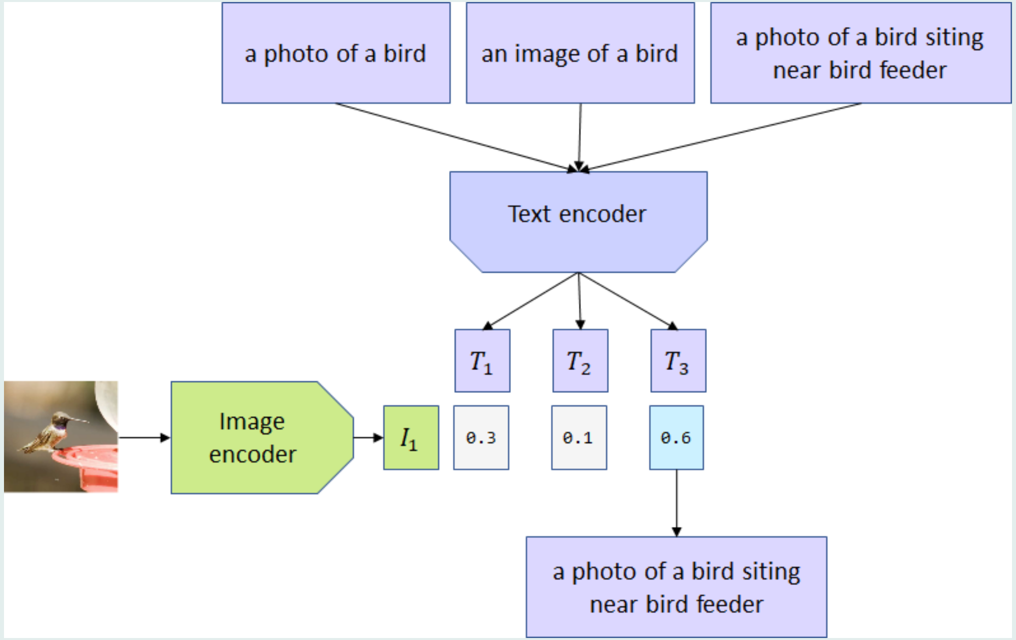
2. CLIPCap


1. Contrastive pre-training
pepper the aussie pup
Text Encoder
Image Encoder
T_1 T_2 T_3 ... T_N
I_1 I_2 I_3 ... I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$
2. Create dataset classifier from label text
plane
car
dog
bird
a photo of a {object}.
Text Encoder
3. Use for zero-shot prediction
Image Encoder
I_1
T_1 T_2 T_3 ... T_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
a photo of a dog.

The CLIPCap model consists of two sub-models, image and text encoders, to encode the input into specific vectors to construct the similarity matrix (I*T is an inner product).

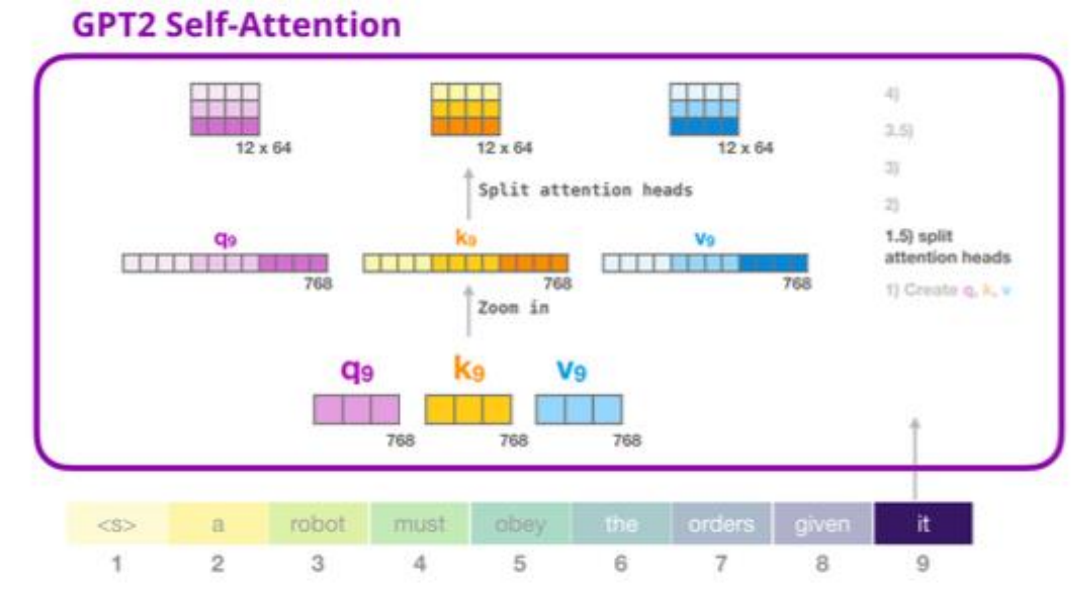
2. CLIPCap

Example



Using pre-train CLIP model (ViT-B/32) to extract semantic information of images
+ Generalization Ability: Outperforms conventional classification models by understanding more about the meaning of image class labels.
+ Usability: The CLIP model has been trained on a huge number of images
Using tokenizer from Hugging Face (*imthanhlv/gpt2news*) to encode caption

2. CLIPCap

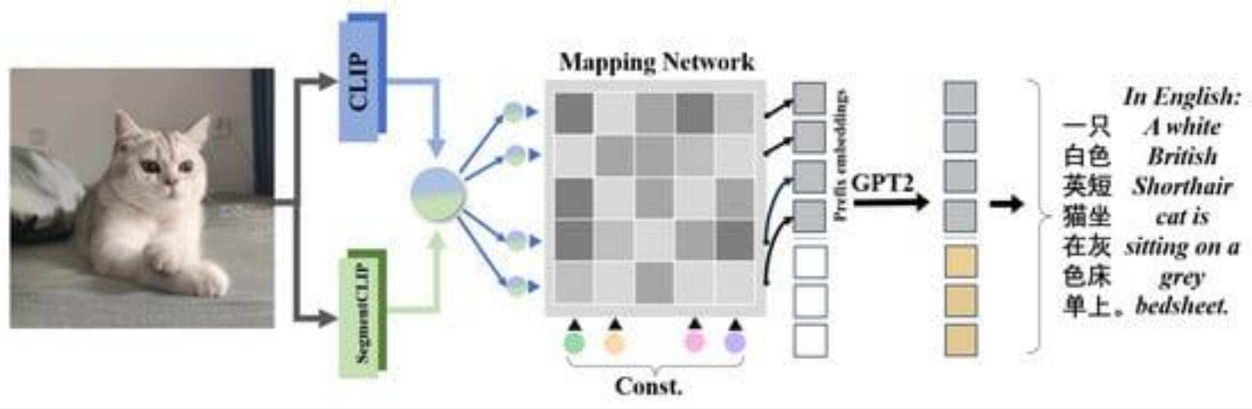


GPT-2 (Generative Pretrained Transformer 2) is a language model based on the Transformer architecture announced by OpenAI in 2019, an unsupervised learning model, trained on the task of predicting the next word in a sentence.

Application in ClipCap: Used for text generation task based on semantic information of CLIP model, to create natural and complete captions.

2. CLIPCap

ClipCap's key – Mapping Network

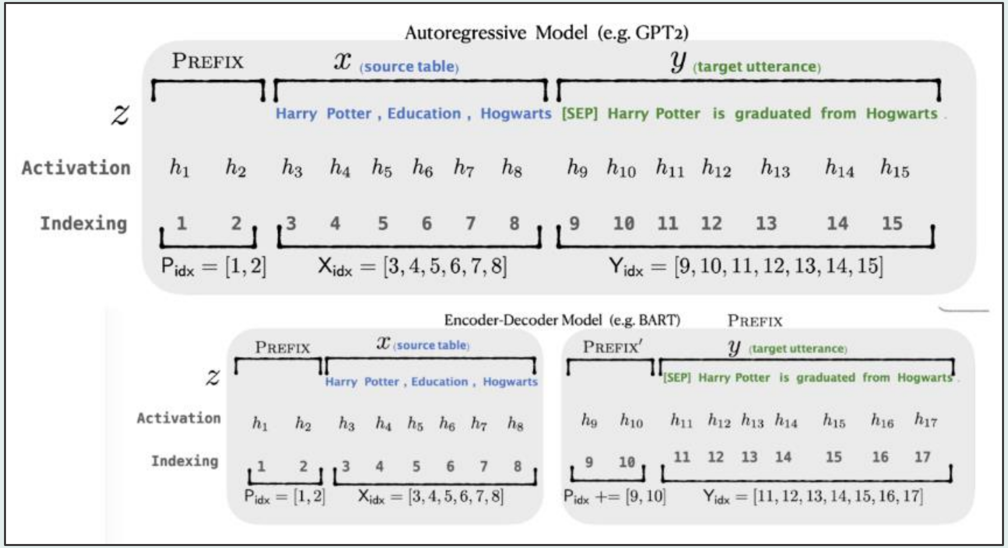


Mapping Network (key component of paper/model) to transform/match CLIP encoding (output of CLIP) to word embedding (input of GPT-2)
Using Prefix-tuning technique for language model, with Transformer network architecture, makes the model significantly lighter,

2. CLIPCap

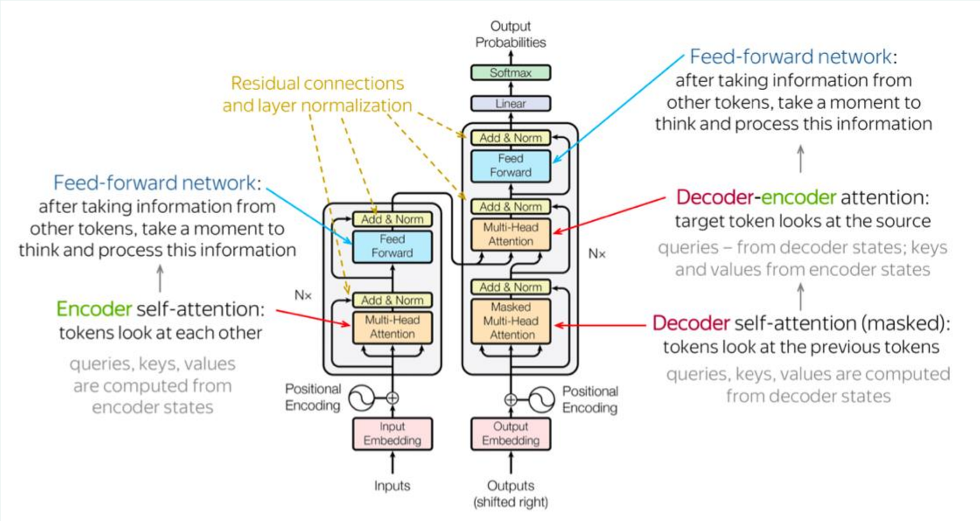
Prefix Tuning and why we use it?

If we want to train the language model to produce a specific target word, “pretraining”/pre-training relevant phrases will strengthen the conditional validation of the language model for the desired output.


Autoregressive Model (e.g. GPT2)
PREFIX
x (source table)
y (target utterance)
z
Harry Potter , Education , Hogwarts
[SEP] Harry Potter is graduated from Hogwarts .
Activation
h_1 h_2 h_3 h_4 h_5 h_6 h_7 h_8 h_9 h_{10} h_{11} h_{12} h_{13} h_{14} h_{15}
Indexing
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
$P_{idx} = [1, 2]$
$X_{idx} = [3, 4, 5, 6, 7, 8]$
$Y_{idx} = [9, 10, 11, 12, 13, 14, 15]$
Encoder-Decoder Model (e.g. BART)
PREFIX
PREFIX
x (source table)
PREFIX'
y (target utterance)
z
Harry Potter , Education , Hogwarts
[SEP] Harry Potter is graduated from Hogwarts .
Activation
h_1 h_2 h_3 h_4 h_5 h_6 h_7 h_8
h_9 h_{10} h_{11} h_{12} h_{13} h_{14} h_{15} h_{16} h_{17}
Indexing
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16 17
$P_{idx} = [1, 2]$
$X_{idx} = [3, 4, 5, 6, 7, 8]$
$P_{idx} += [9, 10]$
$Y_{idx} = [11, 12, 13, 14, 15, 16, 17]$

2. CLIPCap

Transformer



Residual connections and layer normalization

Feed-forward network: after taking information from other tokens, take a moment to think and process this information

Encoder self-attention: tokens look at each other

queries, keys, values are computed from encoder states

Feed-forward network: after taking information from other tokens, take a moment to think and process this information

Decoder-encoder attention: target token looks at the source

queries – from decoder states; keys and values from encoder states

Decoder self-attention (masked): tokens look at the previous tokens

queries, keys, values are computed from decoder states

Minimize the number of words for long sentences.
Transformer includes 2 inputs
- CLIP visual encoding
- Learned constant input: Stores important information learned about CLIP encoding from multi-head attention, helping to optimize the language model when encountering new data.

2. CLIPCap

Using beam search to generate caption:

1. **Beam Initialization:** Generate beam_size beams from the prompt or embedding.
2. **Token Prediction:** The model predicts the next token for each beam and computes log-likelihood scores.
3. **Beam Update:** Select the top beam_size sequences with the highest scores and update the states.
4. **Stopping Check:** Stop the process if all sequences have ended or reached the maximum length.
5. **Return Results:** Return the list of generated sequences, sorted by their scores.

ClipCap Limitations

CLIP will often produce good results on tasks that are recognized across many datasets of common objects, but still struggles with tasks with complex, abstract, or domain-specific objects.

E.g: Task classification on medical dataset, counting number of objects in images, distinguishing different product models, estimating relative distance between objects.

3. ExpansionNet_v2

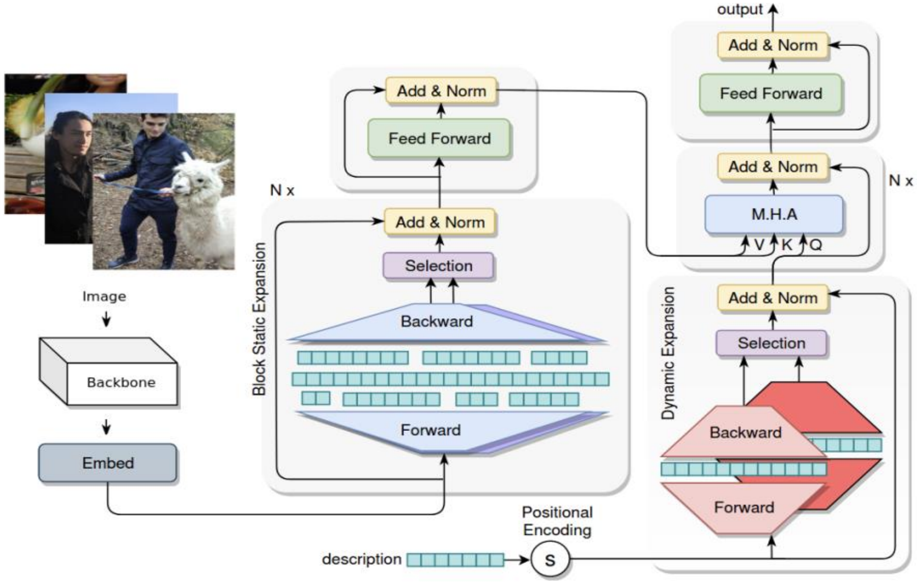
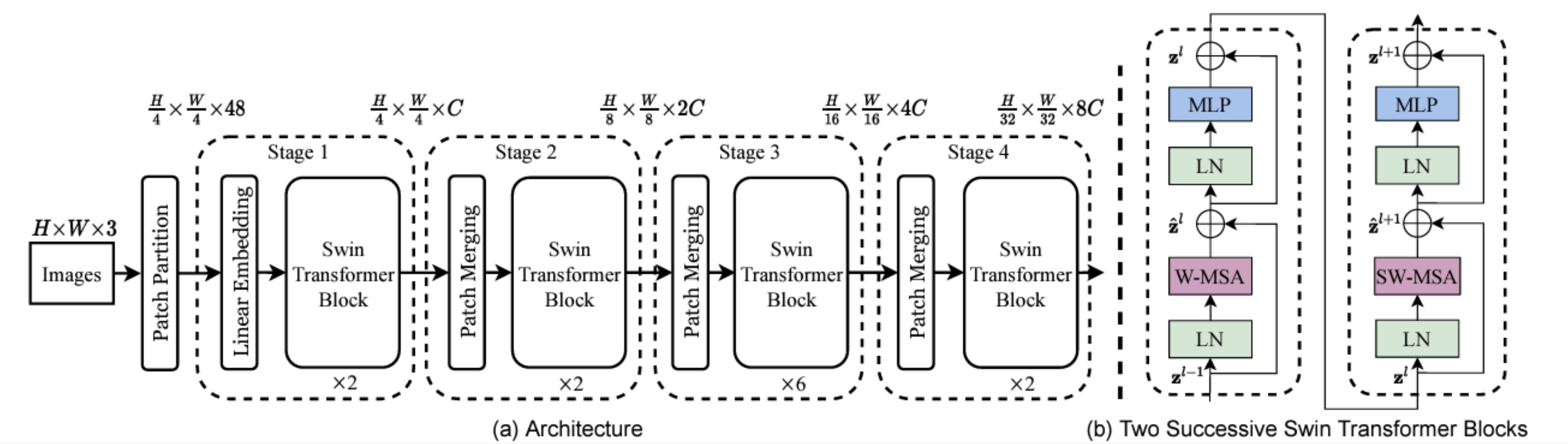


Fig. 3: ExpansionNet v2 architecture.

Swin Transformer



(a) Architecture

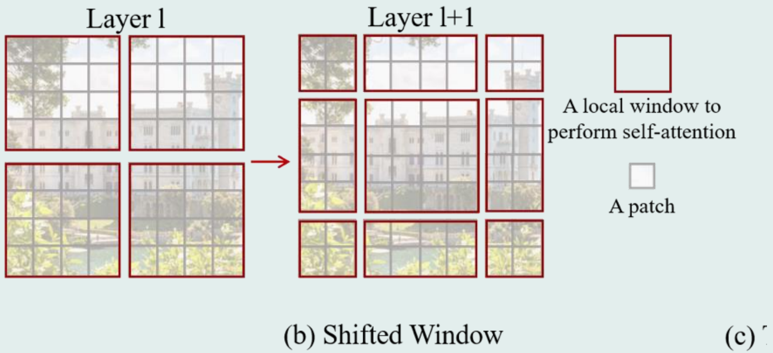
(b) Two Successive Swin Transformer Blocks

Included **4 layers**:

- Layer 1: **Embedding vector creation** with Linear Embedding and Swin Transformer Block.
- Subsequent Layers: Use **Patch Merging** to reduce spatial size and increase vector dimensions, followed by Swin Transformer Block.

Swin Transformer

Swin Transformer uses window-based attention to reduce complexity and shifts the window for interaction between non-adjacent patches.



IV
Experiments & Results

Metrics

NLP-based Metrics:

- **BLEU:** n-gram precision.
- **ROUGE:** Measures longest common subsequence.

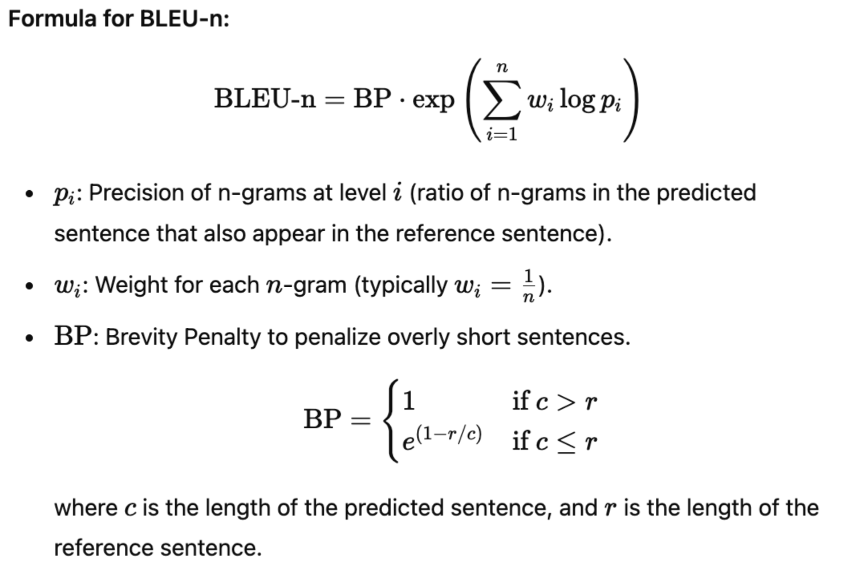
Image-Specific Metrics:

- **CIDEr:** Cosine similarity of TF-IDF weighted n-grams.

1. Metrics

BLEU (Bilingual Evaluation Understudy)

BLEU is a metric based on n-gram overlaps between the predicted sentence and reference sentences.



Formula for BLEU-n:

$$\text{BLEU-n} = \text{BP} \cdot \exp\left(\sum_{i=1}^{n} w_i \log p_i\right)$$

- p_i: Precision of n-grams at level i (ratio of n-grams in the predicted sentence that also appear in the reference sentence).
- w_i: Weight for each n-gram (typically $w_i = \frac{1}{n}$).
- BP: Brevity Penalty to penalize overly short sentences.

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

where c is the length of the predicted sentence, and r is the length of the reference sentence.

BLEU Variants:

BLEU-1: Uses unigram precision (n=1)

BLEU-2: Uses bigram precision (n=2)

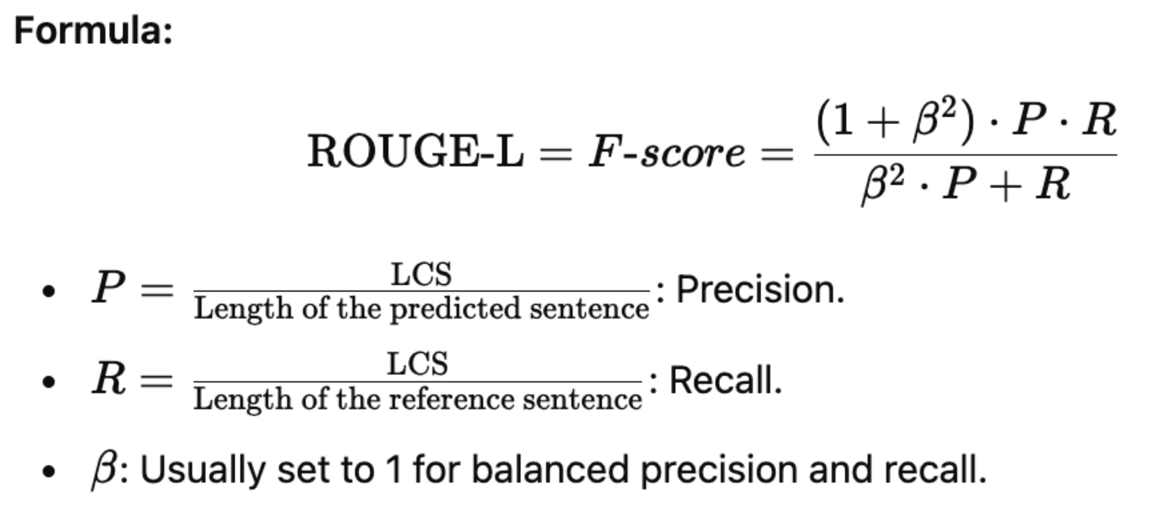
BLEU-3: Uses trigram precision (n=3)

BLEU-4: Uses 4-gram precision (n=4)

1. Metrics

ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation)

ROUGE-L measures the longest common subsequence (LCS) between the predicted and reference sentences.



Formula:

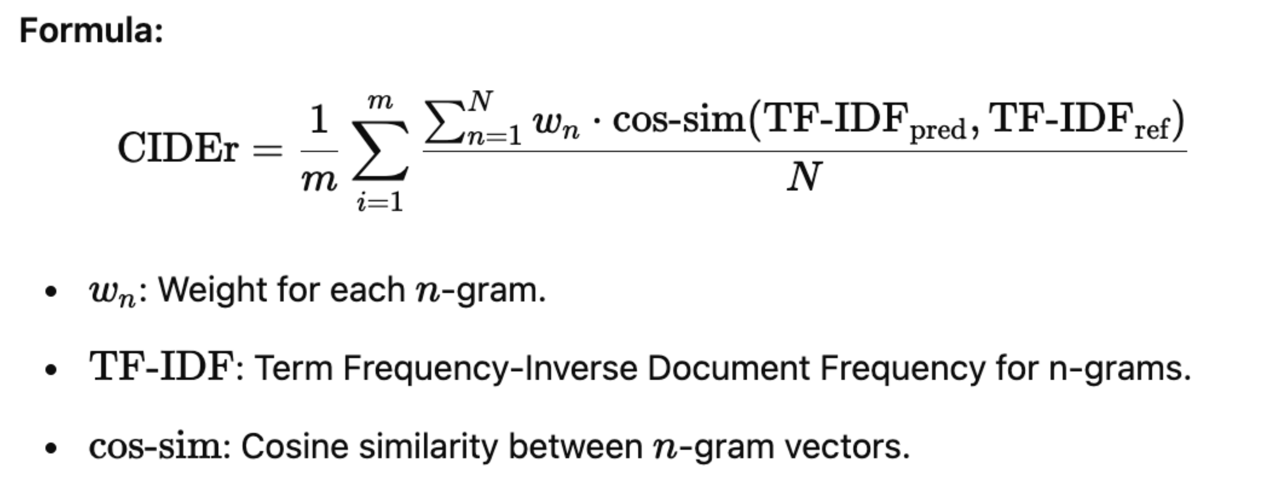
$$\text{ROUGE-L} = F\text{-}score = \frac{(1+\beta^2)\cdot P \cdot R}{\beta^2 \cdot P + R}$$

- $P = \frac{\text{LCS}}{\text{Length of the predicted sentence}}$: Precision.
- $R = \frac{\text{LCS}}{\text{Length of the reference sentence}}$: Recall.
- β: Usually set to 1 for balanced precision and recall.

1. Metrics

CIDEr (Consensus-based Image Description Evaluation)

CIDEr measures the similarity between the predicted and reference sentences using Term Frequency-Inverse Document Frequency (TF-IDF) weighting.



Formula:

$$\text{CIDEr} = \frac{1}{m}\sum_{i=1}^{m}\frac{\sum_{n=1}^{N} w_n \cdot \text{cos-sim}(\text{TF-IDF}_{\text{pred}}, \text{TF-IDF}_{\text{ref}})}{N}$$

- w_n: Weight for each n-gram.
- TF-IDF: Term Frequency-Inverse Document Frequency for n-grams.
- cos-sim: Cosine similarity between n-gram vectors.

05

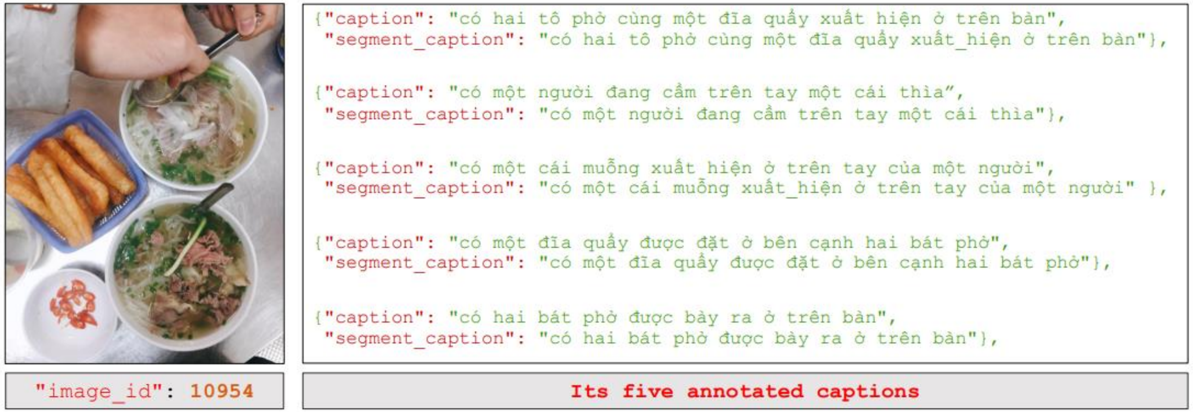
EXPERIMENTS

Dataset

KTVIC Dataset is a comprehensive Vietnamese image captioning dataset centered on daily life activities.

Dataset Details:

+ Images: 4,327
+ Captions: 21,635 in Vietnamese



```
{"caption": "có hai tô phở cùng một đĩa quẩy xuất hiện ở trên bàn",
 "segment_caption": "có hai tô phở cùng một đĩa quẩy xuất_hiện ở trên bàn"},

{"caption": "có một người đang cầm trên tay một cái thìa”,
 "segment_caption": "có một người đang cầm trên tay một cái thìa"},

{"caption": "có một cái muỗng xuất hiện ở trên tay của một người",
 "segment_caption": "có một cái muỗng xuất_hiện ở trên tay của một người" },

{"caption": "có một đĩa quẩy được đặt ở bên cạnh hai bát phở",
 "segment_caption": "có một đĩa quẩy được đặt ở bên cạnh hai bát phở"},

{"caption": "có hai bát phở được bày ra ở trên bàn",
 "segment_caption": "có hai bát phở được bày ra ở trên bàn"},
```

"image_id": 10954

Its five annotated captions

Figure 1: An example of image annotation in the KTVIC dataset, where each image is accompanied by five descriptive (segmented) captions.

Rules

1. Ensure each caption consists of a minimum of 10 Vietnamese words.
2. Only describe visible activities and objects included in the image.
3. Exclude names of places, streets (e.g., Chinatown, New York), and numerical details (e.g., apartment numbers, specific TV times).
4. Allow the use of familiar English words like laptop, TV, tennis, etc.
5. Structure each caption as a single sentence in continuous tense.
6. Omit personal opinions and emotions from annotations.
7. Permit annotators to describe activities and objects from various perspectives.
8. Focus solely on describing visible "thing" objects.
9. Disregard ambiguous "stuff" objects lacking clear borders
10. If there are 10 to 15 objects of the same category or species, annotators may omit them in captions.

Training

Model	Environment	Training parameters	Loss & Optimizer	Time
ExceptionNet_v2	Kaggle (GPU T4x2)	S1: Batchsize : 48, epoch: 8 S2: Batchsize : 16, epoch: 2	Cross Entropy Adam	S1: 3 hours S2: 8 hours
CLIPCap	Kaggle (GPU P100)	ReduceLROnPlateau factor = 0.1 patience = 1 Early stopping patience = 3 50 epochs (stop at 40)		4 hours
EfficientNet_v2 + Transformer	GoogleColab (L4 GPU)	Epoch : 40 Batchsize : 8		1.5 hour

Results

Model	BLEU-4	ROUGE	CIDEr
ExceptionNet_V2	**0.375**	**0.548**	**1.130**
CLIPCap	0.341	0.513	0.944
EfficientNet_v2+ Transformer	0.307	0.483	0.87

06

CONCLUSIONS

Conclusions

- Our team has utilized various deep neural network architectures to generate Vietnamese captions.

- However, our models still face certain limitations such as inaccuracies in counting the number of objects within an image, misidentifying human gender, and other similar challenges.

07

DEMO

Thanks

for listening

References

1. From Show to Tell: A Survey on Deep Learning-based Image Captioning
2. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows
3. Exploiting Multiple Sequence Lengths in Fast End to End Training for Image Captioning
4. KTVIC: A Vietnamese Image captioning Dataset on the Life domain
5. EfficientNetV2: Smaller Models and Faster Training
6. ClipCap: CLIP Prefix for Image Captioning
7. CLIP: Connecting Text and Images
8. CLIP prefix captioning. Github