



Module Code & Module Title

CU6051NI Artificial Intelligence

Assessment Weightage & Type

25% Individual

Semester

2024 Autumn

Project Name: Nepali Image Captioning

Student Name: Miraj Deep Bhandari

London Met ID: 22067814

College ID: np01cp4a220197

Group :L3C8

Assignment Due Date: Wednesday, 25 December 2024

Assignment Submission Date: Monday, 23 December 2024

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

2206784 Miraj Deep Bhandari.docx

 Islington College, Nepal

Document Details

Submission ID

trn:oid::3618:74635077

Submission Date

Dec 22, 2024, 10:05 PM GMT+5:45

Download Date

Dec 22, 2024, 10:08 PM GMT+5:45

File Name

2206784 Miraj Deep Bhandari.docx

File Size

21.8 KB

23 Pages





3,058 Words

17,920 Characters




11% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

-  **27 Not Cited or Quoted 9%**
Matches with neither in-text citation nor quotation marks
-  **5 Missing Quotations 1%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

-  **7% Internet sources**
-  **3% Publications**
-  **9% Submitted works (Student Papers)**

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **27 Not Cited or Quoted 9%**
Matches with neither in-text citation nor quotation marks
- **5 Missing Quotations 1%**
Matches that are still very similar to source material
- **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
- **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 7% Internet sources
- 3% Publications
- 9% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	
	researchrepository.murdoch.edu.au	2%
2	Submitted works	
	University of Greenwich on 2023-01-17	1%
3	Submitted works	
	University College Dublin (UCD) on 2024-06-25	1%
4	Submitted works	
	University of Northampton on 2024-05-02	1%
5	Internet	
	www.coursehero.com	1%
6	Submitted works	
	Bournemouth University on 2023-01-06	1%
7	Internet	
	www.scribd.com	1%
8	Submitted works	
	Liverpool John Moores University on 2021-08-29	1%
9	Internet	
	research.nccgroup.com	0%
10	Submitted works	
	Universidade de Aveiro on 2024-07-31	0%

Table of Contents

1) Introduction:	1
1.1 Aims:	2
1.2 Objectives:	2
1.3 Problem Domain & Use Cases:	3
2) Background:	5
3) Solution:	9
3.1 Algorithms Used in (ViT & GPT):	12
3.2 Pseudocode:	14
3.3 Diagrammatical representations:	17
3.3.1 Image Captioning Model Architecture Diagram:	17
3.3.2 FlowChart Diagram:	18
4) Conclusion:	19
4.1 Analysis of the Work Done:	19
4.2 Addressing Real-World Problems:	20
4.3 Further Work:	20
5) References:	21

Table of Figures

Figure 1: Image Captioning	1
Figure 2: GPT (Decoder)	9
Figure 3: Vision Transformer Architecture (Encoder)	9
Figure 4: Diagram Representation of Entire Architecture	17
Figure 5: Flow Chart Diagram	18

1) Introduction:

Image Captioning is one of the most interesting areas in the field of AI which lies at the intersection of computer vision and natural language processing (NLP). Image captioning is the task of generating a textual description given an image as input. It melds methods for interpreting visual content with the capacity to generate sensible and context-appropriate language.

I am working **on Nepali Image Captioning in Deep Learning** for my project. **Image Captioning** is a task in the **Computer Vision and Natural NLP domain**.

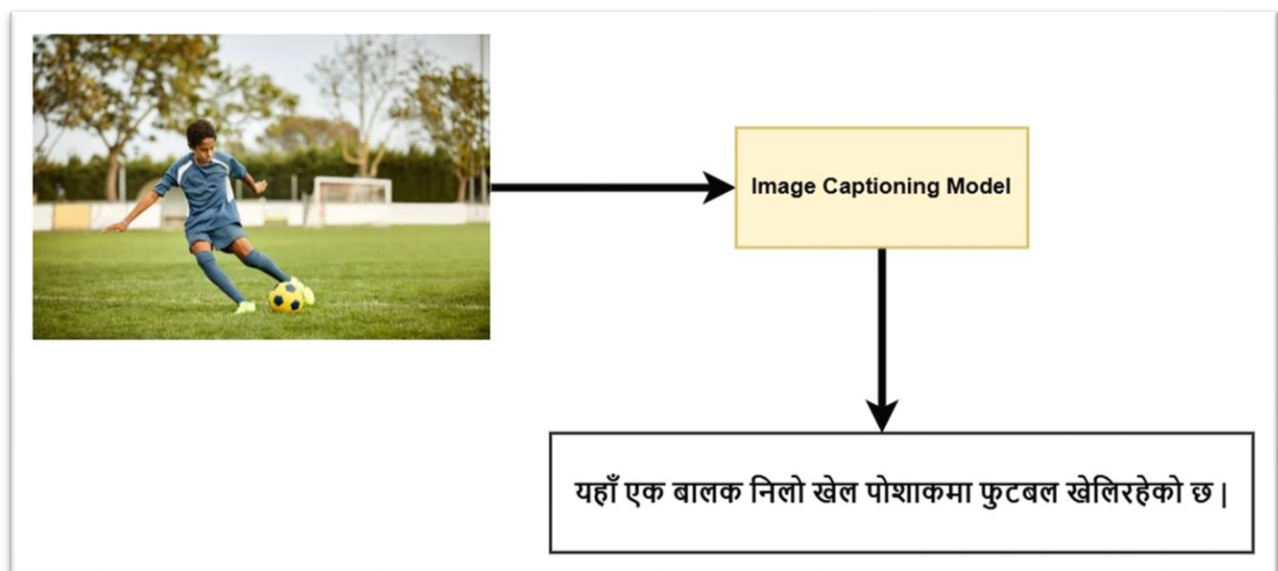


Figure 1: Image Captioning

Computer vision is a branch of AI that allows computers to view and interpret visual data from the surroundings like images and videos. This includes such tasks as: object detection, image classification, segmentation, and creating visual representations, which are all critically important for extracting content from the image in a context of image captioning (Computer Vision for Visual Computing: Techniques and Applications, 1998).

Natural Language Processing(NLP) is the Area of AI to make machines able to communicate in Human Language. Natural language processing techniques are applied to create human-readable captions for the images. This type of task requires knowledge of syntax, background, and semantics so as to generate coherent and determination captions (Rao and McMahan, 2019).

1.1 Aims:

- Create a Deep Learning based system that can Generate the Correct and Corresponding Caption for the Image in the Nepali language.
- Use a combination of the computer vision and the natural language processing to properly interpret the visual content and describe it in natural text.
- Domain-specific linguistic rules and visual content understanding to improve quality of Nepali image captioning.
- Help in the development of AI applications for low resource languages such as Nepali.

1.2 Objectives:

- Use modern Computer Vision algorithms like Vision Transformers for feature extraction from the images.
- Use sequence generation language model such as a transformer to generate Nepali captions from the visual features.
- For training and evaluation of the image captioning model, datasets in Nepali need to be preprocessed and annotated.
- Choose standard metrics (BLEU, METEOR, CIDEr) to evaluate the generated captions for the model.
- Overcome challenges related to grammar, semantics, and contextual appropriateness by Fine-tuning and optimizing the model for Nepali text generation.

1.3 Problem Domain & Use Cases:

Image captioning is essential as it enables machines to describe images using human comprehensible language. Computer vision techniques help us find where these objects are within an image, but they do not tell us what they are in detail. For instance, we can understand when there is a dog in an image, but generating a complete descriptive output like "A dog running on the beach" needs to know actions, context and relations.

Image captioning was invented so that machines can argue with humans in the same space, understand the visual world. It emerged from the need for:

- **Human-Computer Interaction (HCI):** HCI (Human-Computer Interaction): HCI is an approach to computing that styles interactions between humans and machines.
- **Assistive Technologies:** Combining processing visual content and generating accessible descriptions to benefit users with visual impairments.
- **Search and Retrieval:** More searchable image contents through comprehensive captions.
- **Automation and Efficiency:** Automating image organization and product tagging in ecommerce.

Image captioning is useful in several areas:

1. **Social media:** Make automatic image descriptions to improve engagement and accessibility.
2. **E-commerce:** Generating automatic descriptions of products from store images.
3. **Health:** Giving an explanation of medical scans that doctors can better understand.
4. **Autonomous Vehicles:** Assisting self-driving cars in describing what's around them.
5. **Robotics** involves providing robots with the ability to understand their surroundings and make better decisions.
6. **surveillance** : involves the inspection of video streams for the purpose of identifying trends or events.
7. **Education:** Supporting the creation of captions for educational content.
8. **Media and Art:** Automating the selection and composition of content.

2) Background:

In the early years hand-crafted features and rule-based systems dominated the field of image captioning . It was more about retrieving certain visual features from the images and writing captions using predefined templates or simple probabilistic models.

1. Rule-Based Systems (Pre 2000s):

Earlier approaches relied on handcrafted rules to characterize an image. An image, for example, could have been used to detect objects and attributes of an object (color, shape, or position) and then captions could be generated based on a combination of template and attributes.

Example:

Detected objects: “Car,” “Red,” “Road.”

Caption produced: “A red car on the road.”

2. Visual Features and Bag of Words (2000s):

These approaches used manually engineered visual features like **SIFT (Scale-Invariant Feature Transform)** or **HOG (Histogram of Oriented Gradients)** to detect objects or detections in the image.

These were then compared to some fixed collection of labels/captions with simple statistical models based on K-Nearest Neighbors or Bayesian networks.

Limitations:

Must be approved upon design features requiring some domain expertise

Not generalizable at all and unable to do anything more than the base rate

3. Early 2010s: Retrieval-Based Captioning (2000s – Early 2010s):

Such systems relied on having a well labelled database of images and rather than generating captions from scratch, they would find the closest matching image in the database and use it caption.

One example involved matching images using global image descriptors (e.g., GIST) based on their overall similarity.

Limitation:

An inability to generate new captions and generalize with unseen images.

The above methods are the traditional ways of generating captions from images. Modern approaches take a different route in this process with:

1. **Architecture with Encoder and Decoder:** This consists of CNNs (like ResNet) for feature extraction of images and using RNNs (RNNs like LSTMs) to generate captions sequentially.
2. **Attention Mechanisms:** Works by dynamically choosing different relevant parts of the image for each word, instead of calculating one single image per caption. Thus enabling stricter accuracy and detail.
3. **Transformer:** Uses transformers (e.g. Vision Transformers, GPT) for image analysis and caption generation, greater scalability, more fluent, capacity for complex scenes.

For developing the image captioning model, I have gone through the following literature reviews:

Paper Title	Features Implemented	Algorithms Used	Link
Show and Tell: A Neural Image Caption Generator	<ul style="list-style-type: none"> • Uses CNNs to extract features from the image and RNNs to learn to generate a caption describing the image. It uses the encoder-decoder structure with a pre-trained Inception model encoder and a LSTM decoder. 	<ul style="list-style-type: none"> • Convolutional Neural Networks (CNNs) with some famous models like Inception For Feature Extraction. • Long Short Term Memory (LSTM) for sequential language modeling. 	https://arxiv.org/abs/1411.4555
Show, Attend, and Tell: Neural Image Caption Generation with Visual Attention	<ul style="list-style-type: none"> • Adds an attention mechanism which enables the model to pay attention to specific areas of the image when generating each word. This enhances the quality and specificity of the captions generated. 	<ul style="list-style-type: none"> • CNN (ResNet) to extract features. • Whereas a soft attention mechanism focusing on image regions but also introducing a distracting amount of noise. • LSTM for caption generation. 	https://arxiv.org/abs/1502.03044
Image Captioning with Semantic Attention	<ul style="list-style-type: none"> • Semantic attention which fuses global semantic (e.g., latent structured attributes) and spatial visual features. This enhancement helps in the generation of captions which are semantically relevant. 	<ul style="list-style-type: none"> • CNN for extracting spatial features. • Fuse Visual and Semantic features via Semantic Attention Mechanism. • Long short term memory (for generating words one by one). 	https://arxiv.org/abs/1603.03925

<p>Image Captioning with CNN and RNN</p>	<ul style="list-style-type: none"> • It directly investigates the integration of CNN for extracting visual features and RNN for forming the actual sentences. It shows how CNN-based feature vectors can lead the RNN in caption generation. 	<ul style="list-style-type: none"> • Extracting image features using a CNN (VGG16 / ResNet). • Sequentially using GRU (Gated Recurrent Unit) or LSTM to create the caption. 	<p>https://arxiv.org/abs/1604.03944</p>
<p>Image Captioning Using Vision Transformers (ViT)</p>	<ul style="list-style-type: none"> • Suggests using Vision Transformers (ViT) instead of CNNs. Transformers can model the global dependencies of images better than CNNs by getting a larger context. 	<ul style="list-style-type: none"> • Using Vision Transformer (ViT) as feature extractor. • Transformer-based Encoder-Decoder with Transformer for caption generation. • The Multi-Head Attention is used to learn dependencies not only between the image regions but also between the image regions and captions too. 	<p>https://arxiv.org/abs/2105.01928</p>
<p>Attention Is All You Need</p>	<ul style="list-style-type: none"> • Self attention: it captures semantic relationships between all elements in a sequence. • Multi-Head Attention: It allows looking at different places in the input at the same time. • Positional Encoding: Helps in maintaining the order of the sequence. • Encoder-Decoder Architecture: Stacking of both an encoder and a decoder, Encoder-decoder Architecture. 	<ul style="list-style-type: none"> • Scaled Dot-Product Attention. • Layer Normalization, Feedforward Networks. 	<p>https://arxiv.org/abs/1706.03762</p>

3) Solution:

Image captioning can be done using different architectures and techniques. But the latest one is using the **transformer architecture**. The transformer architecture consists of the encoder and decoder parts.

The **encoder** part acts as a **vision transformer**, and the **decoder** part acts as the **GPT-2 architecture**, so overall we can call it **(Vision Transformer + GPT-2) image captioning**. The encoder part takes the images, and it captures the features of the image, and the decoder part, GPT, takes the text with the vision transformer out of the image features and generates the captions of the image.

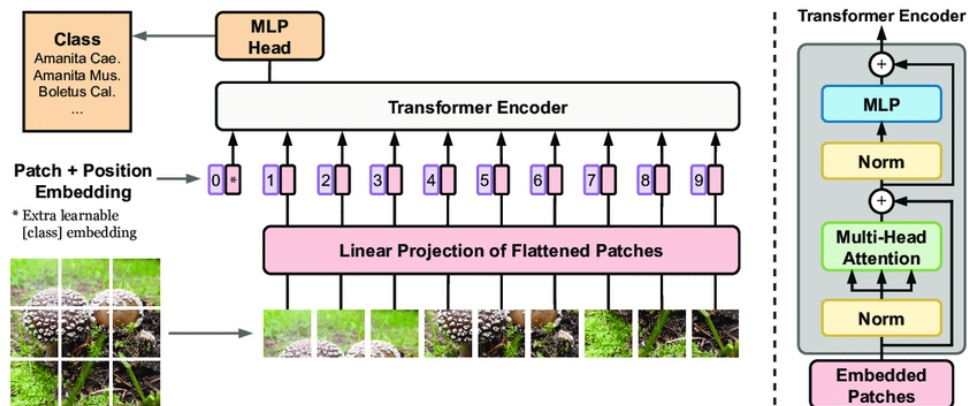


Figure 3: Vision Transformer Architecture (Encoder)

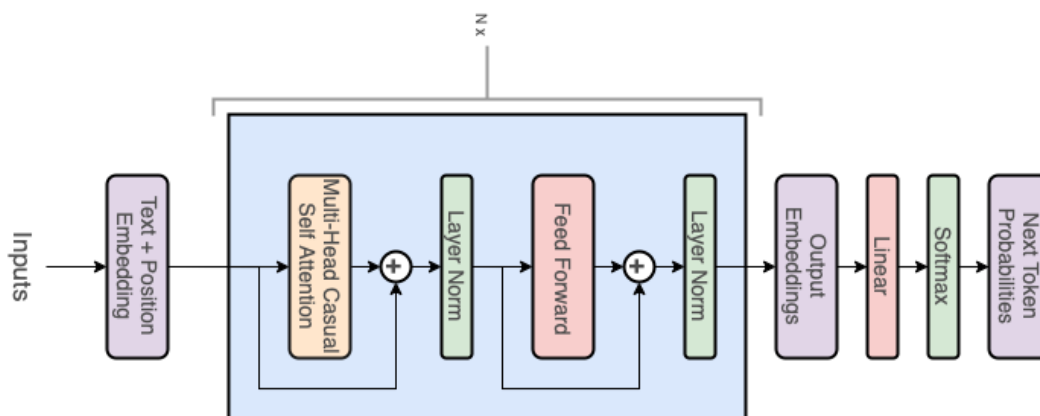


Figure 2: GPT (Decoder)

Image captioning is used for e-commerce, social media like Facebook, scientific research, self-moving vehicles, OCR technologies, medical fields, and many more. The traditional rule and the probabilistic approach don't help to generalize the model. The traditional models are unable to generate the captions for any language; the entire process also involves a lot of time for making the rules and framework. The images are also not processed properly since they consist of heavy matrices, and the bag of words doesn't capture the semantic meaning of words, and the output may always vary for the same image, so the transformer-based models come into play.

The reasons behind the choice of (Vision Transformer + GPT) for image captioning are as follows:

1. Vision Transformer (ViT) for Visual Understanding:

Self-Attention Mechanism : ViT uses self-attention mechanism as their basic process to the image patches. It can capture global dependencies and relationships in the image which act somewhat differently to that of traditional convolutional neural networks (CNNs) which work on local features.

Patch-wise Input: ViT divides the images into fixed-size patches and uses them as tokens (like words in text), which makes it easy to match with the token-level processing of GPT-2, and easy for integration.

Scalability : ViT can easily scale to large datasets and; If we fine-tune on the full ImageNet dataset, ViT takes advantage of pretraining over massive image datasets, yielding strong visual representations for downstream tasks as image captioning and others.

2. GPT for Text Generation:

Autoregressive modeling: As the core functionality behind GPT, autoregressive modeling generates a sequence of words that make the most sense given the previous ones, allowing it to serve as a great caption generator

Pretrained Language Model: Using a pretrained GPT can help generate understandable captions as it also has a lot of linguistic knowledge and fluency in the language.

Fine-Tuning Capability: GPT can be fine-tuned on paired image-caption datasets which enables sophisticated alignment of visual features with the text description.

3. State-of-the-Art Results:

Performance: Combined ViT and GPT-2 often perform better than traditional models since they exploit their strengths as complementary components and achieve state of the art results on image captioning benchmarks.

Generalization: This architecture can generalize on a wide variety of datasets to create captions across various types of images and events.

3.1 Algorithms Used in (ViT & GPT):

The ViT & GPT models consist of various algorithms for image captioning, which are listed below:

1. Positional Encoding:

Positional Encoding injects the positional information into the transformers that handle inputs in a paralleled manner. It does so by simply adding a positional embedding to learnt word vectors that consists of sin and cos of different frequencies, allowing the models to understand the order of tokens.

2. Self-Attention:

Self-Attention enables every token (or image patch) to attend to other tokens in the sequence to build its representation. It is used to capture the semantic meaning between words in sentences. For every token, we convert them to queries, keys, and values vectors. The dot product of the query with all the keys gives the attention scores, which are softmaxed to get weights for the values. It allows the model to relate to each token.

3. Multi-Head Attention:

Multi-Head Attention divides the query, key, and value vectors into separate subspaces for independent processing via different attention heads. The concatenated outputs are reshaped to form a single vector which allows the model to attend to different connections in the input.

4. Masked Attention:

Masked Attention is a specific type of self-attention which masks some positions so that the model cannot attend to future tokens in the sequence. This is very useful for autoregressive tasks such as text generation.

5. Normalization:

We use Layer Normalization to normalize the inputs of a layer by normalizing within the activations of the feature, which has been shown to increase training stability and convergence.

6. Cross-Attention:

Cross-Attention is like self-attention but works over different sequences (i.e. when a decoder attends to the outputs of an encoder in seq-to-seq).

7. Fully Connected Network:

A Fully Connected Network (FCN) is also known as a Dense Network, In this architecture, all the neurons from one layer are connected to all the neurons from the next layer. In this structure, all the nodes of one layer receive input from every node of the previous layer, which makes this a densely interconnected structure.

8. Linear Layer & Softmax:

Linear Layer & Softmax A linear layer projects the final embeddings to logits, and the softmax layer passes these logits to probabilities used in classification or for the generation of outputs.

3.2 Pseudocode:

```
CREATE a class ImageCaptioningSystem

DO

    DECLARE an instance variable imageEncoder as Vision Transformer

    DECLARE an instance variable textDecoder as GPT

    DECLARE an instance variable tokenizer for tokenization and detokenization


CREATE a constructor ImageCaptioningSystem

DO

    INITIALIZE imageEncoder with a pretrained Vision Transformer model

    INITIALIZE textDecoder with a pretrained GPT model

    INITIALIZE tokenizer with the tokenizer compatible with GPT

END DO


CREATE a function preprocessImage

DO

    DECLARE a parameter image

    RESIZE the image to fixed dimensions (e.g., 224x224)

    CONVERT the image to a tensor

    NORMALIZE the image tensor to match Vision Transformer input requirements

    RETURN the preprocessed image

END DO


CREATE a function generateCaption

DO

    DECLARE a parameter image
```

CALL preprocessImage with image and store the result in preprocessedImage

PASS preprocessedImage through imageEncoder to get imageFeatures

INITIALIZE a sequence with a special token <BOS> (beginning of sequence)

WHILE the sequence does not contain <EOS> and the length is less than a maximum

DO

PASS the sequence and imageFeatures to the textDecoder to generate the next token

APPEND the generated token to the sequence

END WHILE

CONVERT the token sequence to text using tokenizer

RETURN the generated caption as a string

END DO

CREATE a function trainModel

DO

LOAD a dataset of image-caption pairs

DECLARE parameters for training such as batch size, learning rate, and epochs

FOR each epoch in total epochs

DO

FOR each batch of image-caption pairs

DO

PREPROCESS images in the batch

TOKENIZE captions in the batch

PASS the images through imageEncoder to get imageFeatures

PASS imageFeatures and tokenized captions through textDecoder for training

CALCULATE the loss between predicted and actual captions

```
        UPDATE model parameters using backpropagation

    END FOR

END FOR

SAVE the trained model

END DO


CREATE a function testModel

DO

    DECLARE a parameter testDataset

    FOR each test image in testDataset

        DO

            CALL generateCaption with the test image

            PRINT the test image and generated caption

        END FOR

    END DO

END DO


CREATE a main function

DO

    DECLARE an input image as inputImage

    CREATE an instance captioningSystem of ImageCaptioningSystem

    CALL captioningSystem.trainModel to train the model

    CALL captioningSystem.generateCaption with inputImage to generate a caption

    DISPLAY the caption

    CALL captioningSystem.testModel with a test dataset to evaluate the model

END DO
```

3.3 Diagrammatical representations:

3.3.1 Image Captioning Model Architecture Diagram:

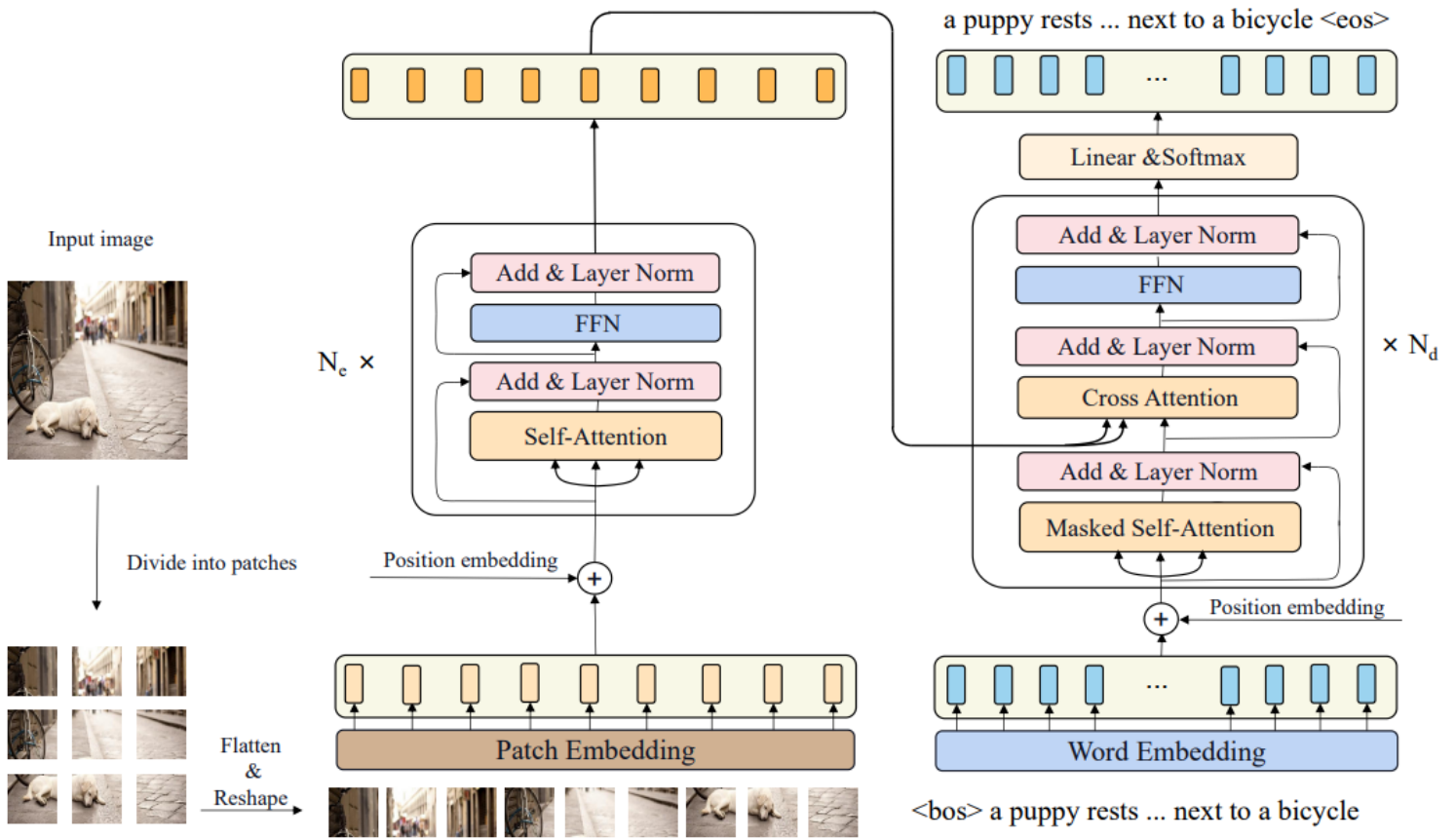


Figure 4: Diagram Representation of Entire Architecture

3.3.2 FlowChart Diagram:

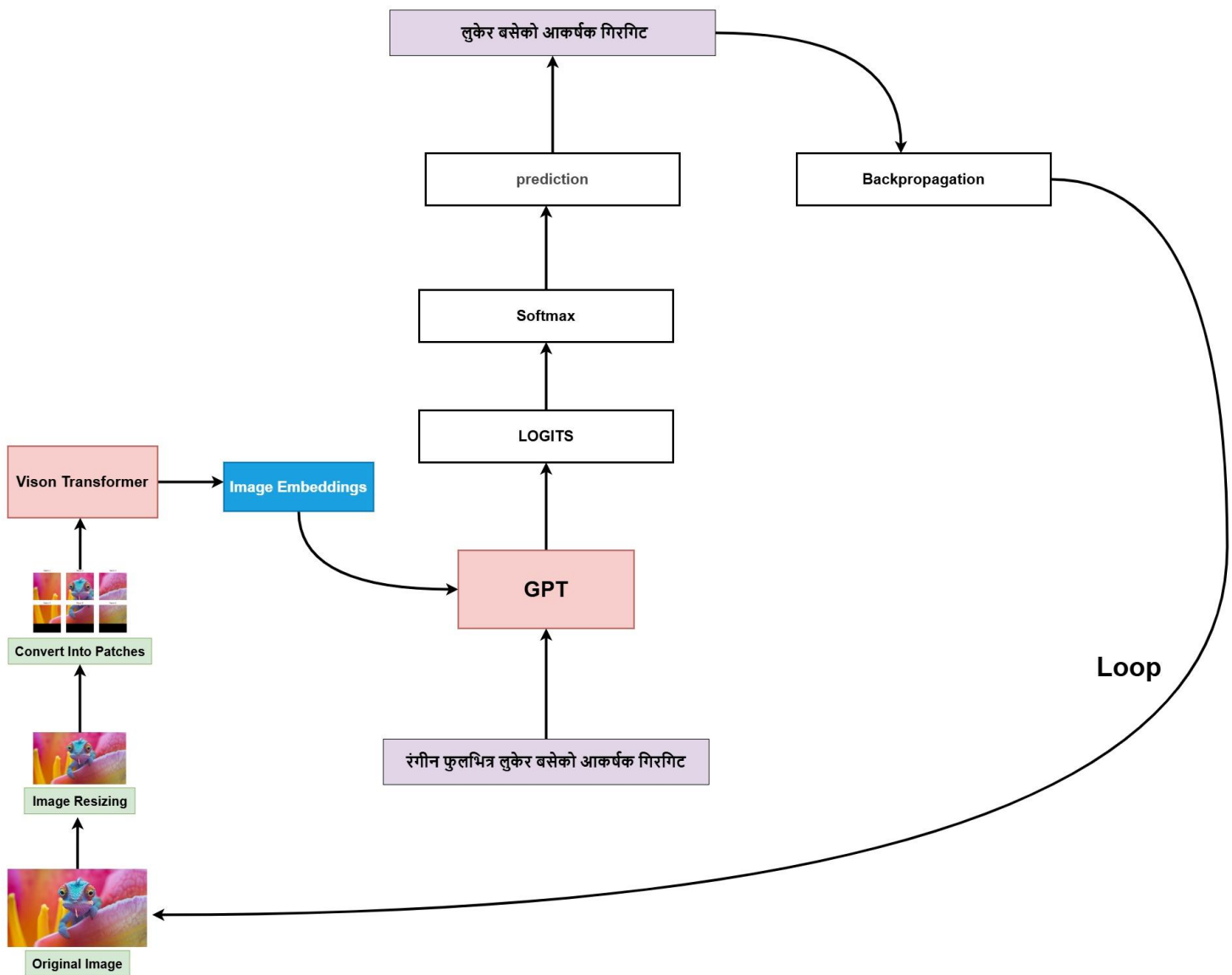


Figure 5: Flow Chart Diagram

4) Conclusion:

This project on Image captioning with Vision Transformers using GPT models is an effective demonstration of how we can combine Machine learning and Deep Learning technologies to create a solution for a real-world problem. By understanding both visual and textual information, the solution produces accurate and contextually relevant captions for a variety of images.

4.1 Analysis of the Work Done:

The implementation build the image captioning model all from scratch, showing good practice in building and training models using deep learning. The Vision Transformers were used for obtaining relevant visual features whereas the text generation part made use of GPT architecture for generating a semantically relevant human-readable description.

To enhance the tokenization process and the generation of captions specifically for the Nepali language, a pre-trained BERT Nepali tokenizer was incorporated. This emphasizes the point that the highly personalized solution was crafted without iterative use of tuned-focused, pre-trained models, and that it reflects new, improved capabilities in model design and performance over the rule-based or probabilistic approaches of the past.

4.2 Addressing Real-World Problems:

This solution supports many applications including:

Visual Impairments: Helping visually impaired users of all kinds by providing relevant image descriptions.

E-commerce: Cataloging and searching the products with product descriptions automation.

Healthcare sector: Used to support the analysis of medical imaging by providing descriptive outputs

Autonomous Systems: Improves perception and decision-making for self-driving cars and robotics.

Social Media: Facebook and Instagram could apply this technology to auto-generate captions for all uploaded images to increase accessibility and user attraction. And automatic captions help people discover your content as well as improve its personalization.

4.3 Further Work:

Further improvements may be done on:

More Multilingual Generating: Scaling up the model to output captions in various languages with increased fluency and accuracy.

Real-time Captioning: Transforming the work done in time-efficient manner so that it can be used in some real-time scenarios.

Multi-Modal Integration: Audio-Visual streams for Multimodal Captioning

Domain Adaptation: Finetuning the model for specific domains, including scientific, surveillance, or educational content.

5) References:

Computer Vision for Visual Computing: Techniques and Applications. (1998). *Computer Vision and Image Understanding*, 71(2), p.153.

doi:<https://doi.org/10.1006/cviu.1998.0714>.

Rao, D. and McMahan, B. (2019). *Natural Language Processing with PyTorch*. 'O'Reilly Media, Inc.'