VIETNAM GENERAL CONFEDERATION OF LABOR
TON DUC THANG UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY

MIDTERM REPORT FOR DEEP LEARNING

MIDTERM REPORT

Instructuor: PhD **LE ANH CUONG**

Writer: **PHAM THIEN VU – 522H0152**

CAO NGUYEN THAI THUAN- 522H0092

Class : **22H50302**

Batch : **26**

HO CHI MINH CITY, YEAR 2025

VIETNAM GENERAL CONFEDERATION OF LABOR

TON DUC THANG UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

MIDTERM REPORT FOR DEEP LEARNING

MIDTERM REPORT

Instructor: PhD **LE ANH CUONG**

Writer: **PHAM THIEN VU – 522H0152**

CAO NGUYEN THAI THUAN- 522H0092

Class : **22H50302**

Batch : **26**

HO CHI MINH CITY, YEAR 2025

THE REPORT WAS COMPLETED AT TON DUC THANG UNIVERSITY

We hereby declare that this report is our own work and has been guided by Dr. Le Anh Cuong. The computational contents, results in this research are genuine and have not been published previously in any form.

If any form of academic dishonesty is found, we take full responsibility for the content of our final report. Ton Duc Thang University is not liable for any copyright infringement or violation caused by me during the execution process (if any)..

Ho Chi Minh city, date month year

Author

*(*sign and write down the name clearly*)*

Pham Thien Vu

Cao Nguyen Thai Thuan

CONFIRMATION AND EVALUATION SECTION BY INSTRUCTOR

Instructor's Confirmation Section

Ho Chi Minh city, date month name
(sign and write down the name clearly)

Instructor's Grading Evaluation Section

Ho Chi Minh city, date month name
(sign and write down the name clearly)

TABLE OF CONTENTS

LIST OF TABLES, FIGURES AND GRAPHS

CHAPTER I – INTRODUCTION

The Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) model is a hybrid deep learning architecture that combines the strengths of Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTMs). Unlike standalone CNNs or LSTMs, this architecture is designed to handle both spatial and temporal data, making it particularly suitable for tasks that require the extraction of spatial features and the modeling of sequential dependencies. The CNN component excels in feature extraction from spatially structured data, such as images or time-series segments, while the LSTM component captures long-term dependencies across sequences.

The applications of CNN-LSTM models are diverse and span multiple fields. In healthcare, they are used for diagnostic predictions based on medical imaging combined with patient history. In energy systems, they forecast time-series energy usage by analyzing historical patterns. In financial markets, they predict trends by integrating structured numerical data with unstructured textual data. This versatility makes CNN-LSTM models highly valuable in scenarios where both spatial and temporal dimensions are critical.

However, real-world data often present challenges such as noise, missing values, and variability in sequence lengths, which can affect the performance of such models. For instance, in time-series forecasting tasks, fluctuations in input data caused by external factors can introduce uncertainty into predictions. Similarly, when dealing with large datasets, the computational complexity of training CNN-LSTM models can become a bottleneck.

To address these challenges, CNN-LSTM models often incorporate techniques like data preprocessing to remove noise, normalization layers to stabilize training, and optimization strategies to reduce computational overhead. Additionally, displaying only the most relevant predictions or insights from the model output is crucial for practical applications where interpretability and efficiency are paramount.

CHAPTER II – ABSTRACT

Chapter's context: This paper is created to solve the problem of answering questions based on images by developing a unified CNN-LSTM model. The approach integrates Convolutional Neural Networks (CNNs) to extract visual features from images and Long Short-Term Memory (LSTM) networks to process textual questions and generate answers. To ensure focused and high-quality data, the scope is limited to a specific domain, such as fruit images, with questions restricted to recognition and quantity. By narrowing the dataset, we aim to enhance data concentration and improve model performance. The paper explores two approaches: using pretrained models and training the model from scratch, while ensuring that both components are seamlessly connected into a unified framework.

2.1 Problem Statement

2.1.1 Unified Image-Question Processing

The CNN-LSTM model is designed to jointly process visual and textual inputs in a single architecture. The CNN component extracts spatial features from images, such as object shapes, colors, or textures, while the LSTM component processes the sequence of words in a question to encode its semantic meaning. These two modalities are fused to generate contextually relevant answers that are conditioned on both the image and the question.

2.1.2 Visual features extraction

The CNN component focuses on learning meaningful features from input images that are essential for answering visual questions. For example, in a dataset of fruit images, the CNN identifies attributes like color, size, and texture that are critical for answering questions such as "How many apples are there?" or "What type of fruit is this ?".

2.1.3 Question understanding

The LSTM component processes natural language questions by encoding their sequential structure and semantic meaning into vector representations. For instance, given a question like "What color is the fruit?", the LSTM captures key elements such as "color" and "fruit" to guide answer generation.

2.2 The Given Input/ Expected Output

To solve this problem effectively, we will implement two approaches: using pretrained CNN models such as ResNet-50 combined with LSTM layers for efficient feature extraction, and training a CNN-LSTM model from scratch tailored specifically to this task. These approaches will be evaluated based on accuracy in generating correct answers, semantic coherence in responses, computational efficiency in terms of runtime and memory usage during training, and robustness across diverse image-question pairs.

Metrics such as precision, recall, F-score for answer prediction tasks will be used alongside qualitative assessments for semantic coherence to determine which method provides the most effective solution for answering questions about images using a unified CNN-LSTM architecture.

CHAPTER III – PRELIMINARIES

Chapter's context: To help solve the problem of answering questions based on images using a CNN-LSTM model, this section introduces several key terms and concepts. These preliminaries provide the theoretical foundation for understanding the components, methodologies, and challenges of building a unified architecture that processes both visual and textual data.

3.1 High Utility Itemset Mining

Convolutional Neural Networks (CNNs) are deep learning models designed to process structured spatial data, such as images. CNNs use convolutional layers to extract features like edges, textures, and shapes by applying filters across input images. These features are critical for understanding visual content and are particularly useful in tasks like object recognition and classification. For example, in our task, CNNs identify attributes such as object shape or color from an image to support answering visual questions.

3.2 Features Ex

Feature extraction in CNNs involves identifying patterns within images that are relevant to the task at hand. Filters in convolutional layers detect local features such as edges or textures, while pooling layers reduce dimensionality to retain only significant information. For instance, given an image of a red apple, a CNN can extract features like its round shape and red color. These features are then passed to the LSTM component for further processing .

3.3 Incremental Database

Pretrained CNN models like ResNet-50 leverage learned features from large datasets such as ImageNet to reduce training time while maintaining high accuracy. For example, ResNet-50 can be used to extract high-level features from fruit images, such as texture or shape variations between apples and oranges, before passing these features to the LSTM component for generating answers.

3.4 Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed to process sequential data while addressing issues like vanishing gradients. LSTMs use memory cells and gates (input, output, and forget gates) to selectively retain or discard information across time steps. In this task, LSTMs process textual questions word by word to understand their semantic meaning and generate answers based on both image features and question context.

3.5 Sequential Processing

LSTMs excel at handling sequential data by maintaining dependencies between inputs over time steps.

For example, given a question like “How many apples are there?”, the LSTM processes each word sequentially while retaining contextual information about “apples” and “quantity”.

3.6 Semantic Encoding

Semantic encoding converts natural language questions into vector representations that capture their meaning.

For instance, an LSTM encodes the question "What type of fruit is this?" into a representation that emphasizes keywords like "type" and "fruit," enabling the model to generate an accurate answer based on extracted image features.

3.7 CNN - LSTM Integration

The integration of CNNs and LSTMs forms a unified architecture capable of processing both spatial and temporal data simultaneously. The CNN component extracts spatial features from images, which are then passed as input to the LSTM component alongside encoded textual questions.

3.8 Features Fusion

Feature fusion combines spatial features extracted by CNNs with sequential representations generated by LSTMs into a single framework. For example, in answering “How many apples are there?” from an image of apples, the CNN identifies the number of objects while the LSTM interprets the query to generate the answer “Five”.

3.9 Questions and Answers Genertaion

Questions and answers are generated using LLMs based on collected images to ensure consistency and relevance within the dataset.

For example: Image:

- A photo of three bananas.
- Question: "How many bananas are there?"
- Answer: "Three."

CHAPTER IV – PROPOSED METHODOLOGY

4.1 Theoretical basis

4.1.1 Context

In this chapter, we explore the theoretical foundation of the CNN-LSTM connected model, which is designed to solve the problem of answering questions based on images. The model combines Convolutional Neural Networks (CNNs) for spatial feature extraction and Long Short-Term Memory (LSTM) networks for sequential question processing into a unified architecture. The goal is to create a system capable of understanding visual and textual inputs simultaneously, generating answers conditioned on both modalities.

The CNN component processes the input image to extract spatial features such as object shapes, colors, and textures. These features are then passed to the LSTM component, which encodes the question into a semantic representation by processing it word by word. The integration of these two components is achieved through an attention mechanism that aligns relevant image regions with the encoded question vector, ensuring that the generated answer is contextually accurate.

4.1.2 CNN-LSTM Connected Algorithm

The algorithm begins with image feature extraction using a CNN backbone, such as ResNet-50. The CNN processes the input image through multiple convolutional layers, extracting hierarchical features that represent various aspects of the image. For example, given an image of apples, the CNN identifies attributes like their shape, color, and texture while ignoring irrelevant background details. Simultaneously, the question is tokenized and passed through an embedding layer to convert each word into a dense vector representation. These embeddings are processed by a bidirectional LSTM, which captures both forward and backward dependencies in the question. For instance, for the question "How many apples are there?", the LSTM encodes key elements such as "how many" and "apples" into a fixed-size vector that represents the semantic meaning of the entire question. The attention mechanism plays a critical role in aligning these two modalities. It computes relevance scores for each region of the image based on its alignment with the encoded question vector. Using these scores, it generates a context vector by weighting image features according to their relevance to the question. For example, if the question asks about "apples," attention focuses on regions in the image containing apples while ignoring other areas. The context vector from attention is fused with the encoded question vector in an encoder fusion layer to create a "thought vector." This thought vector encapsulates both visual and textual information relevant to answering the question. It is then passed to an LSTM decoder that generates answers one token at a time. During training, teacher forcing is used to improve convergence by feeding ground truth tokens as input at each step. During inference, however, the decoder uses its own predictions as input for subsequent steps until an end-of-sequence (<EOS>) token is generated or a maximum length is reached.

4.2 Implementation in a practical environment

In this section, we will investigate how the given pseudocodes in our selected papers are implemented in practice.

4.2.1 CNN-LSTM Integration Algorithm

4.2.1.1 Pseudocode

CLASS VQAModel:

INITIALIZE(vocab_size, embedding_dim=300, hidden_dim=512, pretrained=False):

DEFINE vocab_size ← vocab_size

DEFINE special tokens: sos_idx ← <SOS>, eos_idx ← <EOS>, pad_idx ← <PAD>

INITIALIZE CNN encoder:

```
        cnn ← ResNet50(weights=None)
    INITIALIZE question encoder:
        embedding ← Embedding(vocab_size, embedding_dim)
        question_encoder ← Bidirectional LSTM(embedding_dim, hidden_dim)
        question_projection ← Linear(hidden_dim*2 → hidden_dim)
    INITIALIZE attention mechanism:
        attention ← Attention(image_dim=512, question_dim=hidden_dim)
    INITIALIZE encoder fusion layer:
        encoder_fusion ← Sequential(
            Linear(512 + hidden_dim → hidden_dim),
            ReLU(),
            Dropout(0.5)
        )
    INITIALIZE decoder:
        decoder ← LSTM(embedding_dim → hidden_dim)
        output_projection ← Linear(hidden_dim → vocab_size)
FUNCTION ENCODE(image, question, question_lengths):
    PROCESS image through CNN layers:
        img_features ← CNN(image)  # Extract spatial features

    RESHAPE img_features for attention:
        img_features ← RESHAPE(img_features → [batch_size, num_regions, feature_dim])
    PROCESS question through embedding and LSTM:
        embedded_question ← EMBEDDING(question)
        packed_question ← PACK-PADDED-SEQUENCE(embedded_question, question_lengths)
```

_, (hidden_states, cell_states) ← LSTM(packed_question)

COMBINE bidirectional outputs:

question_features ← CONCAT(hidden_states[-2], hidden_states[-1])

question_features ← LINEAR-PROJECTION(question_features)

APPLY attention mechanism:

context_vector, _ ← ATTENTION(img_features, question_features)

FUSE features into thought vector:

thought_vector ← LINEAR-FUSION(CONCAT(context_vector, question_features))

RETURN thought_vector, cell_states[-1]

FUNCTION FORWARD(image, question, question_lengths, answer=None, teacher_forcing_ratio=0.5):

ENCODE image and question:

thought_vector, memory_cell ← ENCODE(image, question, question_lengths)

INITIALIZE decoder input with <SOS> token:

decoder_input ← FULL([batch_size], value=<SOS>)

INITIALIZE decoder hidden state with encoded thought vector and memory cell:

decoder_hidden_state ← (EXPAND-DIM(thought_vector), EXPAND-DIM(memory_cell))

PREPARE output tensor for predictions:

target_length ← LENGTH(answer) if answer ≠ None else MAX_LENGTH

outputs ← ZEROS([batch_size, target_length, vocab_size])

FOR t in range(0 to target_length):

EMBED current input token:

```
            decoder_embedded_input ← EMBEDDING(decoder_input)
          PROCESS through decoder LSTM:
            decoder_output, decoder_hidden_state ← LSTM(decoder_embedded_input, decoder_hidden_state)
          PROJECT output to vocabulary probabilities:
            prediction ← LINEAR-PROJECTION(decoder_output.squeeze(1))
            outputs[:, t] ← prediction
          DETERMINE next input token (teacher forcing or model prediction):
            use_teacher_forcing ← RANDOM() < teacher_forcing_ratio AND answer ≠ None
            IF use_teacher_forcing THEN
              decoder_input ← answer[:, t]
            ELSE
              top_indices ← TOP-K(prediction)[0]
              decoder_input ← top_indices
              IF ALL(decoder_input == <EOS>) THEN
                BREAK
        RETURN outputs
      FUNCTION GENERATE-ANSWER(image, question, question_lengths):
        ENCODE image and question:
          thought_vector, memory_cell ← ENCODE(image, question, question_lengths)
        INITIALIZE decoder input with <SOS> token:
          decoder_input ← FULL([batch_size], value=<SOS>)

        INITIALIZE decoder hidden state with encoded thought vector and memory cell:
```

```
      decoder_hidden_state ← (EXPAND-DIM(thought_vector), EXPAND-DIM(memory_cell))
    STORE generated tokens:
      generated_tokens = []
    FOR step in range(0 to MAX_LENGTH):
      EMBED current input token:
        decoder_embedded_input = EMBEDDING(decoder_input)
      PROCESS through decoder LSTM:
        decoder_output, decoder_hidden_state = LSTM(decoder_embedded_input, decoder_hidden_state)
      PROJECT output to vocabulary probabilities:
        prediction = LINEAR-PROJECTION(decoder_output.squeeze(1))
      GET most likely next token:
        top_indices = TOP-K(prediction)[0]
        token = top_indices.item()
        generated_tokens.append(token)
      BREAK if <EOS> generated:
        IF token == <EOS> THEN
          BREAK
        SET next input as current prediction:
          decoder_input = top_indices
    RETURN generated_tokens
END CLASS
```

4.2.1.2 Operations

The CNN-LSTM integrated algorithm operates as a unified framework for processing both visual and textual inputs to generate answers based on the given image and question. The process begins with the CNN encoder, which extracts spatial features from the input image by passing it through convolutional layers, pooling operations, and residual blocks. These features are reshaped into a format suitable for further processing, treating each region of the image as a separate token. Concurrently, the textual question is tokenized and embedded into dense vector representations before being processed by a bidirectional LSTM. This LSTM captures both forward and backward dependencies within the question, producing a semantic vector that represents its overall meaning. The attention mechanism dynamically aligns relevant regions of the image with the encoded question vector, generating a context vector that highlights regions most pertinent to answering the question. This context vector is fused with the encoded question representation to form a "thought vector," which encapsulates both visual and textual information. The thought vector is passed to an LSTM decoder, which generates tokens sequentially to form the answer. During training, teacher forcing improves convergence by feeding ground truth tokens as input, while inference relies on predictions to generate subsequent tokens until an end-of-sequence (<EOS>) token is produced or a maximum length is reached. The efficiency of this algorithm lies in its seamless integration of spatial and sequential data, leveraging pretrained CNNs for robust feature extraction and bidirectional LSTMs for precise semantic encoding.

4.2.2 Attention Algorithm

4.2.2.1 Pseudocode

```
CLASS Attention:
    INITIALIZE(image_dim=512, question_dim=512, attention_dim=512):
        DEFINE projection layers for image and question features:
            image_projection = Linear(image_dim → attention_dim)
            question_projection = Linear(question_dim → attention_dim)
        DEFINE attention vector for computing weights:
            attention_vector = Linear(attention_dim → 1)
    FUNCTION FORWARD(image_features, question_features):
```

EXPAND dimensions of question features to match spatial regions of image features:

expanded_question_features = REPEAT-DIM(question_features.unsqueeze(1), num_regions=image_features.size(1))

PROJECT both image and question features to common attention space:

img_proj = IMAGE-PROJECTION(image_features)

ques_proj = QUESTION-PROJECTION(expanded_question_features)

COMPUTE joint attention features using tanh activation:

joint_attention = TANH(img_proj + ques_proj)

CALCULATE attention scores using learned weights:

attention_scores = ATTENTION-VECTOR(joint_attention).squeeze(-1)

APPLY softmax to normalize scores into attention weights:

attention_weights = SOFTMAX(attention_scores)

COMPUTE context vector as weighted sum of image features based on attention weights:

context_vector = SUM(attention_weights.unsqueeze(-1) * image_features along dimension=1)

RETURN context_vector, attention_weights

END CLASS

4.2.2.2 Operations

The Attention algorithm dynamically aligns visual features extracted from the image with the encoded semantics of the question to ensure relevance in answer generation. It begins by projecting both image features and question features into a shared attention space using linear transformations. Each region of the image is scored based on its relevance to the question using learned attention weights, calculated through joint feature interactions in this shared space. These scores are normalized via a softmax function to produce attention weights that determine how much focus each region should receive. The weighted sum of image features is computed using these attention weights, resulting in a context vector that emphasizes regions most relevant to answering the question. For example, if the question asks "How many apples are there?", attention focuses on regions containing apples while ignoring irrelevant areas like shadows or background objects. This context vector is returned alongside attention weights, enabling interpretability by highlighting which parts of the image contributed most to answering the question. The efficiency of this algorithm lies in its ability to prioritize relevant visual information based on textual input, reducing computational overhead by focusing only on significant regions rather than processing all image data equally.

CHAPTER V – RESULTS AND ANALYSIS

Chapter's context: In this chapter, we will demonstrate the results of these proposed algorithms after running with testing dataset. Here is the result of our finding.

5.1 Accuracy testing

To find out the accuracy of our algorithms among each other, we will be using these datasets :

breed	image_path	question	answer	question_type
Afghan	valid/Afghan/01.jpg	What grooming requirements are specific to a A	The Afghan is unique for its blend of gentle yet alert traits	general
Afghan	valid/Afghan/02.jpg	What are the defining physical features of a Afg	A versatile breed, the Afghan combines adaptability with	general
Afghan	valid/Afghan/03.jpg	How does the Afghan's temperament make it un	This breed exhibits a stubborn personality, thriving in sub	temperament
Afghan	valid/Afghan/04.jpg	What makes a Afghan suitable or unsuitable for	Afghans are known for their intelligence and unwavering	general
Afghan	valid/Afghan/05.jpg	What unique challenges come with training a Af	A versatile breed, the Afghan combines adaptability with	general
Afghan	valid/Afghan/06.jpg	What are the most recognizable personality trait	With a high-energy disposition, the Afghan fits well with	temperament
Afghan	valid/Afghan/07.jpg	What grooming requirements are specific to a A	The Afghan is unique for its blend of agile and curious trai	general
Afghan	valid/Afghan/08.jpg	What makes a Afghan suitable for families?	The Afghan is best for experienced owners for family life	suitability
Afghan	valid/Afghan/09.jpg	How does the Afghan's coat affect its care needs	This breed stands out due to its 28 inches, 25 lbs, and broa	appearance
Afghan	valid/Afghan/10.jpg	What are the historical origins of the Afghan bre	Historically, the Afghan was known for palace guard in cer	history
African Wild Dog	valid/African Wild Dog/01.jpg	What makes a African Wild Dog suitable for fami	The African Wild Dog is moderately compatible for family	suitability
African Wild Dog	valid/African Wild Dog/02.jpg	How does the African Wild Dog's coat affect its c	African Wild Dogs are recognized by their medium, golder	appearance
African Wild Dog	valid/African Wild Dog/03.jpg	How does the African Wild Dog's temperament r	With a high-energy disposition, the African Wild Dog fits v	temperament
African Wild Dog	valid/African Wild Dog/04.jpg	What are the defining physical features of a Afri	African Wild Dogs are known for their intelligence and pla	general
African Wild Dog	valid/African Wild Dog/05.jpg	What unique challenges come with training a Af	This breed stands out due to its gentle disposition and cor	general
African Wild Dog	valid/African Wild Dog/06.jpg	How much exercise does a African Wild Dog nee	African Wild Dogs need a balanced diet and consistent joi	care
African Wild Dog	valid/African Wild Dog/07.jpg	How does a African Wild Dog perform in dog spo	Historically, African Wild Dogs have performed well in tra	performance
African Wild Dog	valid/African Wild Dog/08.jpg	What are the historical origins of the African Wil	Historically, the African Wild Dog was known for palace gu	history
African Wild Dog	valid/African Wild Dog/09.jpg	What makes a African Wild Dog suitable or unsui	A versatile breed, the African Wild Dog combines adaptab	general
African Wild Dog	valid/African Wild Dog/10.jpg	What are the defining physical features of a Afri	African Wild Dogs are known for their endurance and play	general
Airedale	valid/Airedale/01.jpg	What unique challenges come with training a Ai	Airedales are known for their endurance and unwavering	general
Airedale	valid/Airedale/02.jpg	What are the historical origins of the Airedale br	The Airedale traces its origins to England where it was use	history
Airedale	valid/Airedale/03.jpg	How does the Airedale's coat affect its care nee	A defining characteristic of the Airedale is its double-laye	appearance
Airedale	valid/Airedale/04.jpg	What are the defining physical features of a Aire	This breed stands out due to its gentle disposition and tra	general
Airedale	valid/Airedale/05.jpg	What are the defining physical features of a Aire	This breed stands out due to its keen senses and guarding	general
Airedale	valid/Airedale/06.jpg	How does the Airedale's temperament make it u	With a calm and patient disposition, the Airedale fits well	temperament
Airedale	valid/Airedale/07.jpg	What grooming requirements are specific to a Ai	Airedales are known for their intelligence and playful ene	general
Airedale	valid/Airedale/08.jpg	What unique challenges come with training a Ai	Airedales are known for their loyalty and playful energy.	general

Figure 5.1: Validate dataset.

With this dataset, we will observe the results from running the algorithm with two different option, with pretrained model and housetrained model to find out which one is the most consistent and have higher accuracy. The dataset have a total of 700 entries, with 5 columns to test the computational of the algorithm. The results are as follows:

```
--------------------------------------------------
Epoch 39/70, Train Loss: 0.6044, Val Loss: 5.8071, BLEU: 0.0952
Question: how does a african wild dog perform in dog sports or work roles?
Target: historically, african wild dogs have performed well in tracking trials.
Predicted: this african wild valued the wild dog is suited for tasks.
--------------------------------------------------
Question: what makes a african wild dog suitable or unsuitable for first-time owners?
Target: a versatile breed, the african wild dog combines adaptability with keen intelligence.
Predicted: a versatile breed, the african wild dog combines gentleness with keen intelligence.
--------------------------------------------------
Question: how does the african wild dog's coat affect its care needs?
Target: african wild dogs are recognized by their medium, golden, and their curly tail.
Predicted: the african wild are is by their large, black and tan, and their curly tail.
--------------------------------------------------
Epoch 40/70, Train Loss: 0.6042, Val Loss: 5.7463, BLEU: 0.0962
Question: how does a african wild dog perform in dog sports or work roles?
Target: historically, african wild dogs have performed well in tracking trials.
Predicted: with african wild speed, the wild dog is suited for tasks.
--------------------------------------------------
Question: what makes a african wild dog suitable or unsuitable for first-time owners?
Target: a versatile breed, the african wild dog combines adaptability with keen intelligence.
Predicted: a african dogs are known for their intelligence and intuitive energy.
--------------------------------------------------
Question: how does the african wild dog's coat affect its care needs?
Target: african wild dogs are recognized by their medium, golden, and their curly tail.
Predicted: the african wild are is for their large, black tan, and their curly tail.
--------------------------------------------------
Epoch 41/70, Train Loss: 0.5870, Val Loss: 5.8716, BLEU: 0.1009
Question: how does a african wild dog perform in dog sports or work roles?
Target: historically, african wild dogs have performed well in tracking trials.
Predicted: historically, african wild dogs have well its and is known for strength.
--------------------------------------------------
```

Figure 5.2: Truncated outputs of the pretrained model.

```
Predicted: the african dogs dog is famous its smooth coat, which is low maintenance. needs.
--------------------------------------------------
Epoch 39/70, Train Loss: 0.4957, Val Loss: 6.1483
Question: how does a african wild dog perform in dog sports or work roles?
Target: historically, african wild dogs have performed well in tracking trials.
Predicted: with its wild have performed in skills and is known for intelligence.
--------------------------------------------------
Question: what makes a african wild dog suitable or unsuitable for first-time owners?
Target: a versatile breed, the african wild dog combines adaptability with keen intelligence.
Predicted: a versatile breed, out african wild dog combines with keen intelligence.
--------------------------------------------------
Question: how does the african wild dog's coat affect its care needs?
Target: african wild dogs are recognized by their medium, golden, and their curly tail.
Predicted: the african dogs are of african wild dog is its fur and striped coat.
--------------------------------------------------
Epoch 40/70, Train Loss: 0.4897, Val Loss: 5.9641
Question: how does a african wild dog perform in dog sports or work roles?
Target: historically, african wild dogs have performed well in tracking trials.
Predicted: with african sprint speed, the wild dog is suited for tasks.
--------------------------------------------------
Question: what makes a african wild dog suitable or unsuitable for first-time owners?
Target: a versatile breed, the african wild dog combines adaptability with keen intelligence.
Predicted: a versatile wild are known for their intelligence and intuitive
--------------------------------------------------
Question: how does the african wild dog's coat affect its care needs?
Target: african wild dogs are recognized by their medium, golden, and their curly tail.
Predicted: the african dogs are recognized their large, black and tan, and curly tail.
--------------------------------------------------
Epoch 41/70, Train Loss: 0.4958, Val Loss: 6.1481
Question: how does a african wild dog perform in dog sports or work roles?
Target: historically, african wild dogs have performed well in tracking trials.
Predicted: with its wild dogs performed well in obedience trials.
--------------------------------------------------
```

Figure 5.3: Truncated outputs of the housetrained model.

With these results, we can see that all the pretrained model has signifincantly higher word by word accuracy and the sentences were more meaningful.

5.2 Time efficiency

Pre-trained models demonstrate significantly faster training times compared to models trained from scratch. This efficiency stems from the fact that pre-trained models have already learned generalizable features from large datasets (like ImageNet), establishing optimized weight parameters. When fine-tuning on new data, these models begin from an advantageous position in the parameter space, requiring only adaptations to domain-specific features rather than learning fundamental visual representations from random initialization. The optimization process converges more rapidly because the model leverages previously learned patterns, effectively transferring knowledge from the pre-training domain to the target task. This transfer learning approach substantially reduces the number of iterations needed to reach target performance metrics compared to training with randomly initialized weights.

CHAPTER VI – CONCLUSION

Chapter's context: In this chapter, we will make some final remarks about the proposed algorithm while also discuss about some future measures that can be taken to improve on the algorithms.

After delving into the problem and the proposed methodologies to solve the problem, we can conclude that for a bigger dataset, we might achieve better overall results.

In the future, we can further improve these algorithms by implementing some practices to make sure that the code is easier to comprehend. Another possible solution that can be taken is utilizing a different data structure that focus more on accessibility and time efficiency.

REFERENCE MATERIALS

English

1. Aris, T. N. M., Ningning, C., Mustapha, N., & Zolkepli, M. (2024). Integration of CNN and LSTM Networks for Behavior Feature Recognition: An Analysis. International Journal on Advanced Science, Engineering and Information Technology, 14(5), 1793-1799.
2. Abdallah, M., Khac, N. A. L., Jahromi, H., & Jurcut, A. D. (2021). A Hybrid CNN-LSTM Based Approach for Anomaly Detection Systems in SDNs. Proceedings of the 16th International Conference on Availability, Reliability and Security (ARES 2021).
3. Mijanur Rahman, P. (2022). Different ways to combine CNN and LSTM networks for time series classification tasks, 257.

JOB DIVISION TABLE

Student name	Problem and literature analysis	Code implementation	Report contribution	Task completion(%)
Cao Nguyen Thai Thuan	50%	30%	70%	100%
Pham Thien Vu	50%	70%	30%	100%