Case Study Intelligent Systems

Image caption generator

Group members:

Samant Vinayak - 22303846

Chaudhary Akash – 12305919

Besharatil Ali -22302913

Tom Emil - -22310012

Under the guidance of

Prof. Ginu Paul

Contents

- Introduction
- Problem Statement & Background
- Objectives & Goal
- Literature Review
- Dataset Used
- Model 1
- Model 2
- Model Evaluation & Comparison
- Realworld Application
- Future Scope and conclusion

Introduction

Can machines truly understand what they see?

Significance

Accessibility: Automated image descriptions can assist visually impaired individuals by providing them with verbal descriptions of visual content.

Content Management: It can enhance the organization and retrieval of images in large databases by generating descriptive metadata automatically.

Social Media: Automatic captions can improve user experience by generating relevant descriptions for images shared on social platforms, aiding in content discovery and engagement.

E-commerce: Enhanced product descriptions for images can improve searchability and user experience on online shopping platforms.

Problem Statement & Background

Problem Statement-

- The intelligent system aims to address the problem of automatic image caption generation
- The model created must create meaningful and coherent textual descriptions for images without human intervention
- The challenge lies in accurately interpreting visual content and generating relevant, grammatically correct sentences that describe the image comprehensively

Background-

- With the explosion of visual data on the internet and social media, there is a growing need for systems that can understand and describe images automatically.
- Convolutional Neural Networks (CNN) for image recognition and Long Short-Term Memory (LSTM) networks for sequence prediction, have made image captioning more advanced.
- Able to create models to generate captions by understanding the context and content of images with thier deep learning techniques.

Specific Objective and Goals

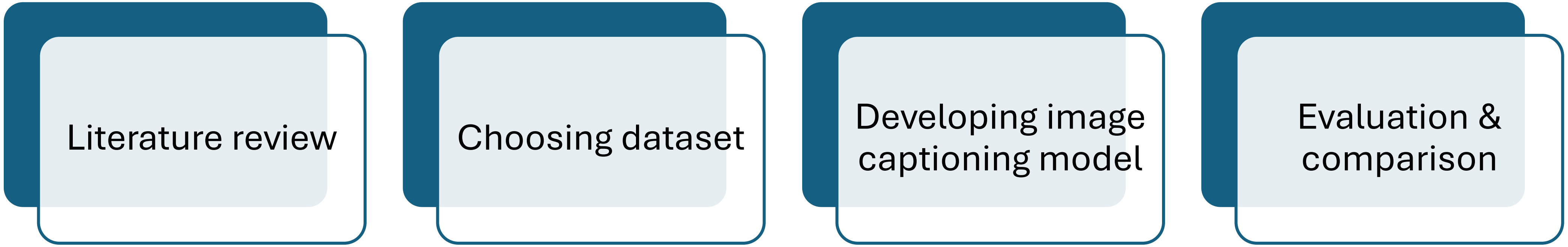
Accurate Image Understanding

To generate grammatically correct, coherent, and contextually relevant captions

To create a system that performs well across a wide range of image types

Real-Time Processing

Task Description



Literature review

Choosing dataset

Developing image captioning model

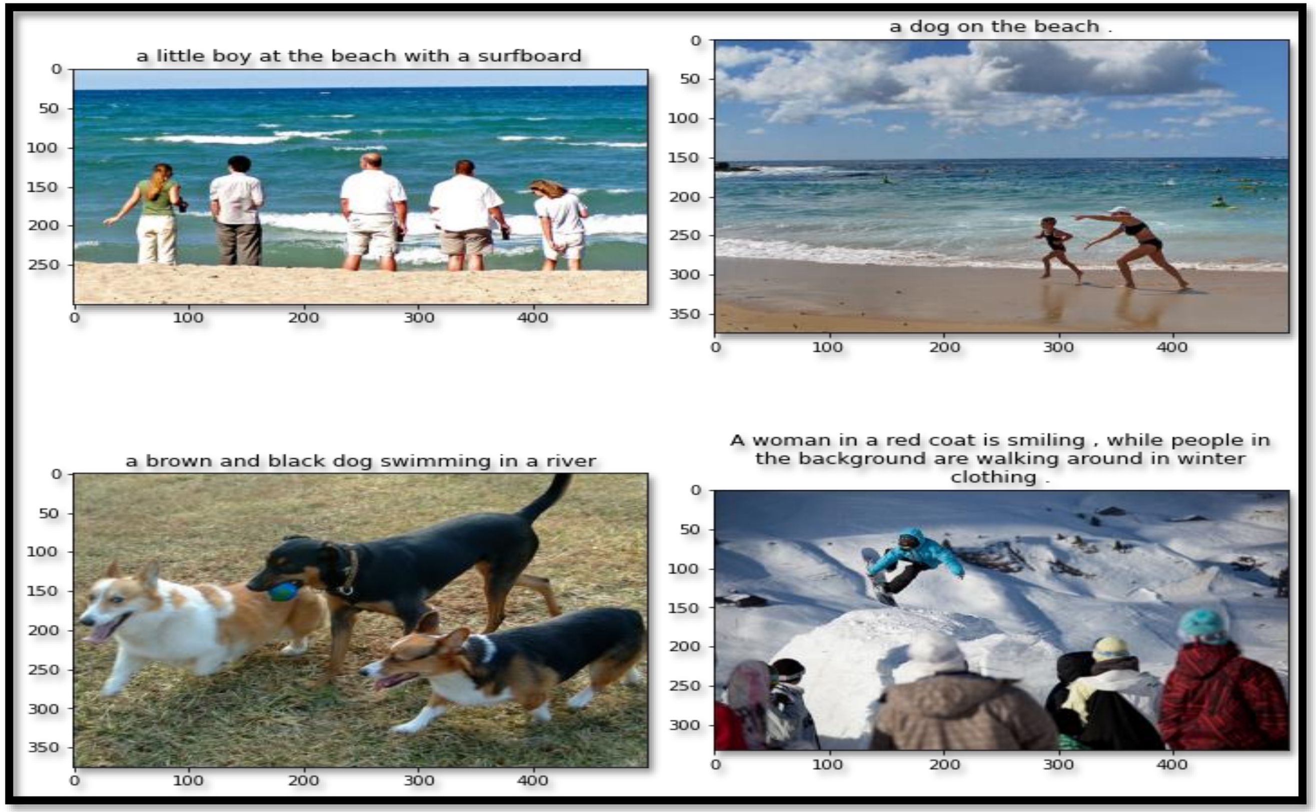
Evaluation & comparison

Literature review

No	Paper Title	Author	Outcome
1	Transformative Fusion: Vision Transformers and GPT-2 Unleashing New Frontiers in Image Captioning within Image Processing	Indrani Vasireddy, G. HimaBindu, and Ratnamala.B	The paper presents a approach that combines **Vision Transformers and GPT-2** for image captioning and aims for accurate descriptions of visual content
2	Python Based Project to Build Image Caption Generator With CNN & LSTM	Kanishk Tawde and Akshit Garg	This paper presents a Python-based project that systematically analyzes the utilization of **CNN and LSTM** networks for generating descriptive captions from images,
3	Show and Tell: A Neural Image Caption Generator	Oriol Vinyals,Alexander Toshev, Samy Bengio and Dumitru Erhan	This paper introduces NIC, a **neural network** system that automa tically generates descriptive sentences for images by encoding them with a convolutional neural network and ge nerating corresponding sentences wit h a **recurrent neural network**

Dataset

- flickr
- comes in 2 files one for images and other for texts/captions
- Examples of diverse activities from a Flickr dataset

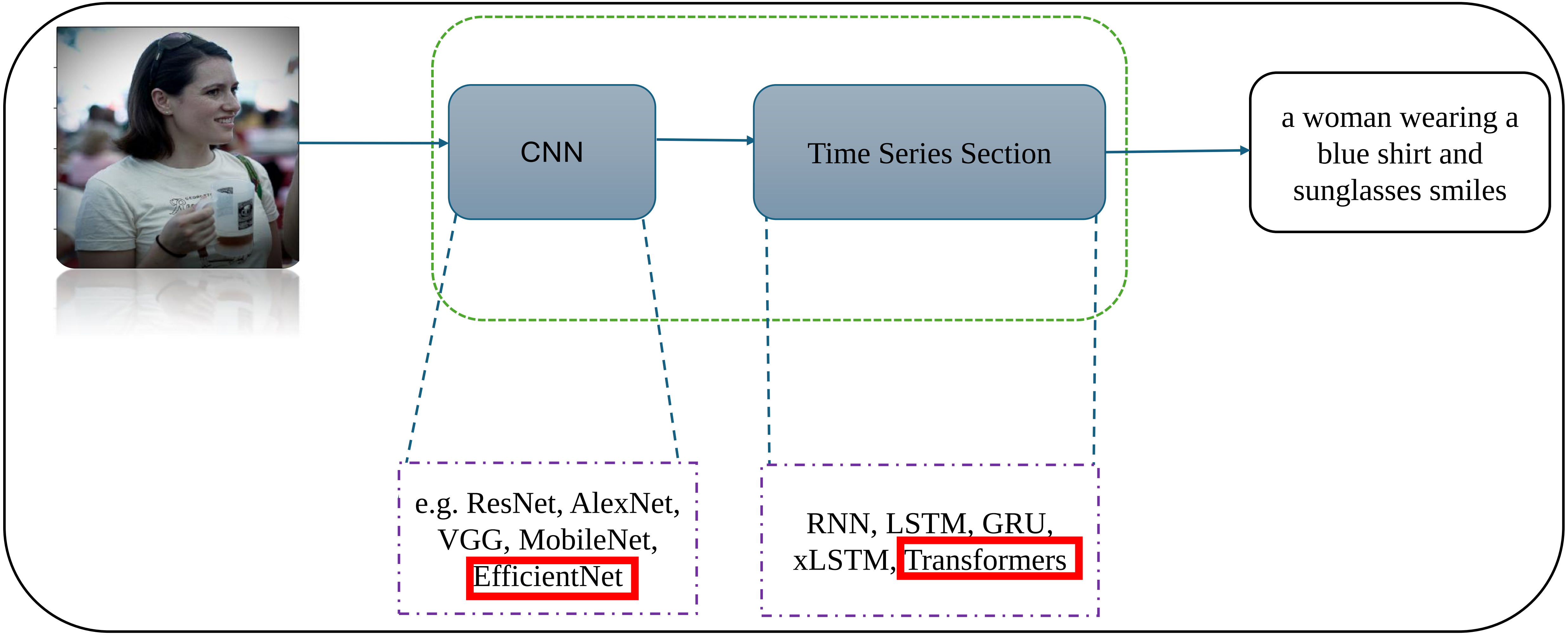

a little boy at the beach with a surfboard

a dog on the beach .

a brown and black dog swimming in a river

A woman in a red coat is smiling , while people in the background are walking around in winter clothing .

Caption Model 1



CNN

Time Series Section

a woman wearing a blue shirt and sunglasses smiles

e.g. ResNet, AlexNet, VGG, MobileNet, EfficientNet

RNN, LSTM, GRU, xLSTM, Transformers

Transformers

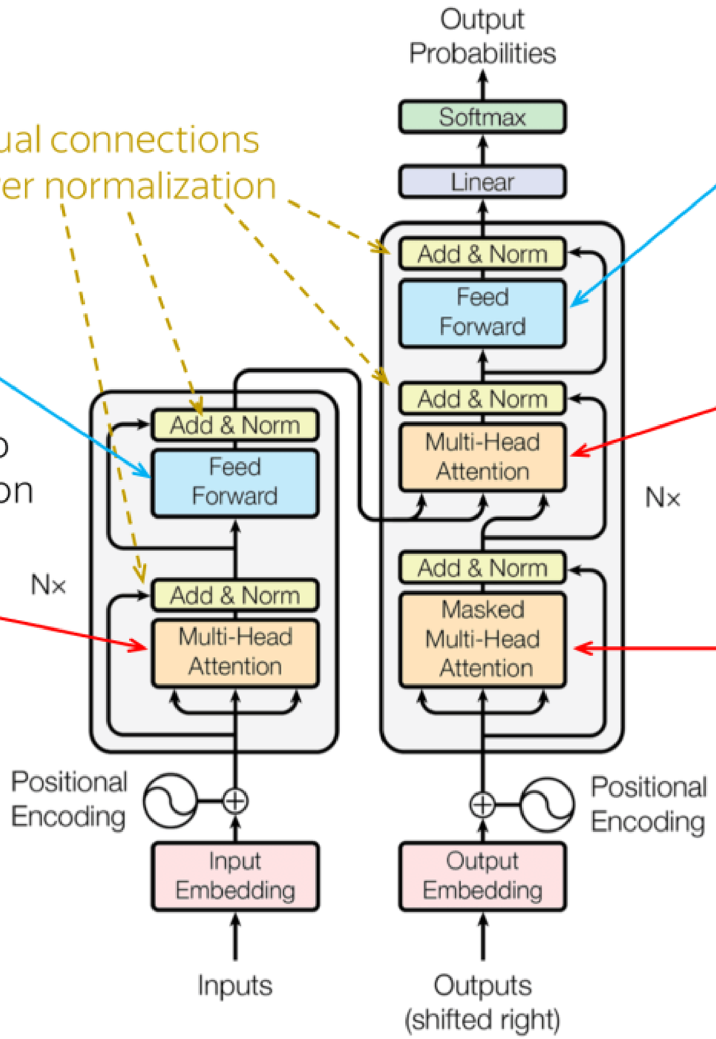
Residual connections and layer normalization

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

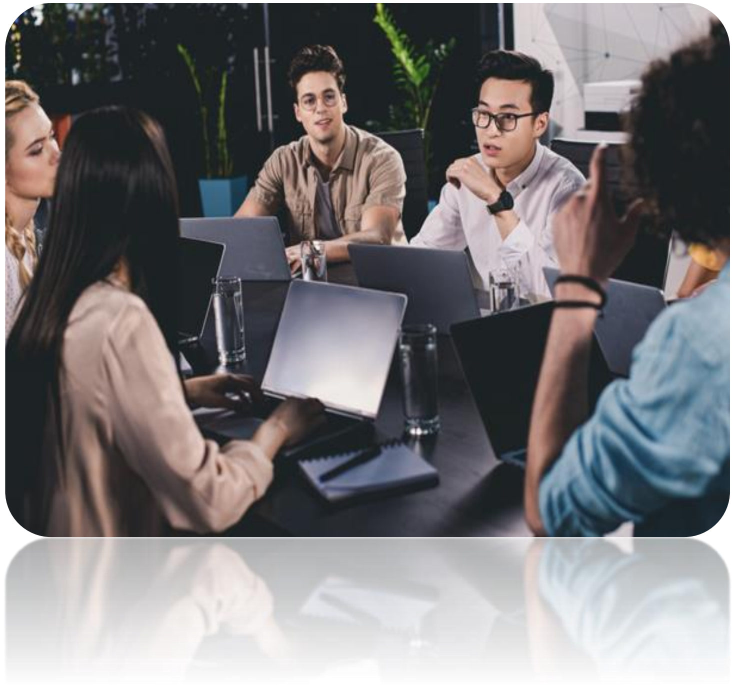
queries – from decoder states; keys and values from encoder states

↑

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

queries, keys, values are computed from decoder states

Testing on new samples – Unseen data



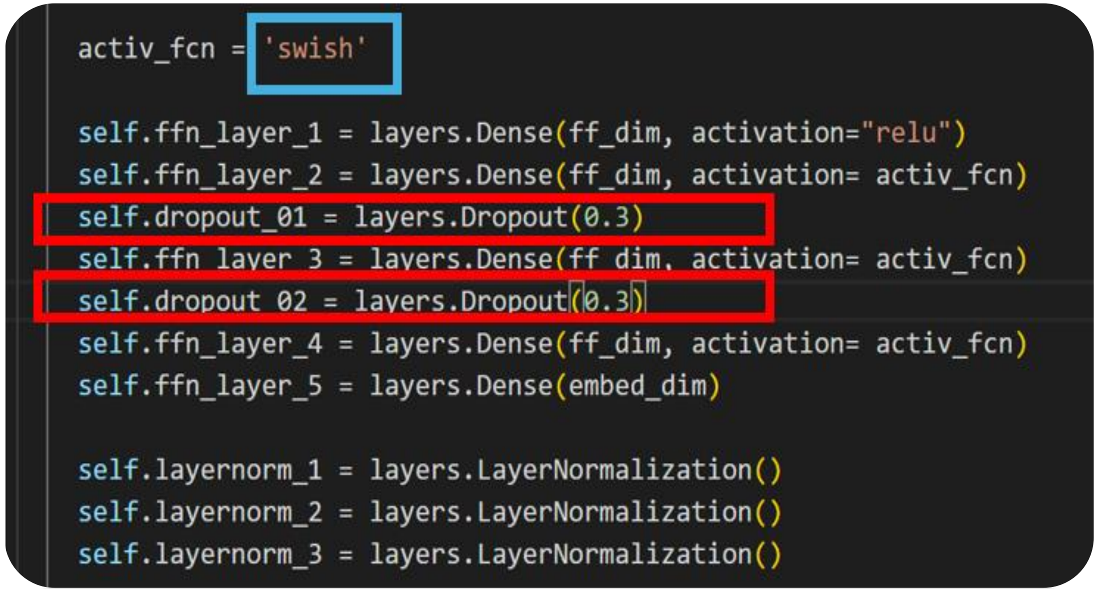
CNN Model	Caption
ResNet50	a man and a woman are sitting in a white chair
VGG19	a man in a white shirt and a woman in a white shirt and a black shirt and white shirt is holding a man
InceptionV3	a man in a red shirt and white shorts is standing on a sidewalk
ResNet50V2	a man in a black shirt and white shirt is holding a baby in a blue cup
MobileNetV2	a man in a blue shirt and blue jeans is standing in front of a crowd
EfficientNetB0	a man and a woman are sitting at a table with a table
EfficientNetB7	a man in a blue shirt is holding a baby in a large glass doors

Training Results and Metrics of Different Models

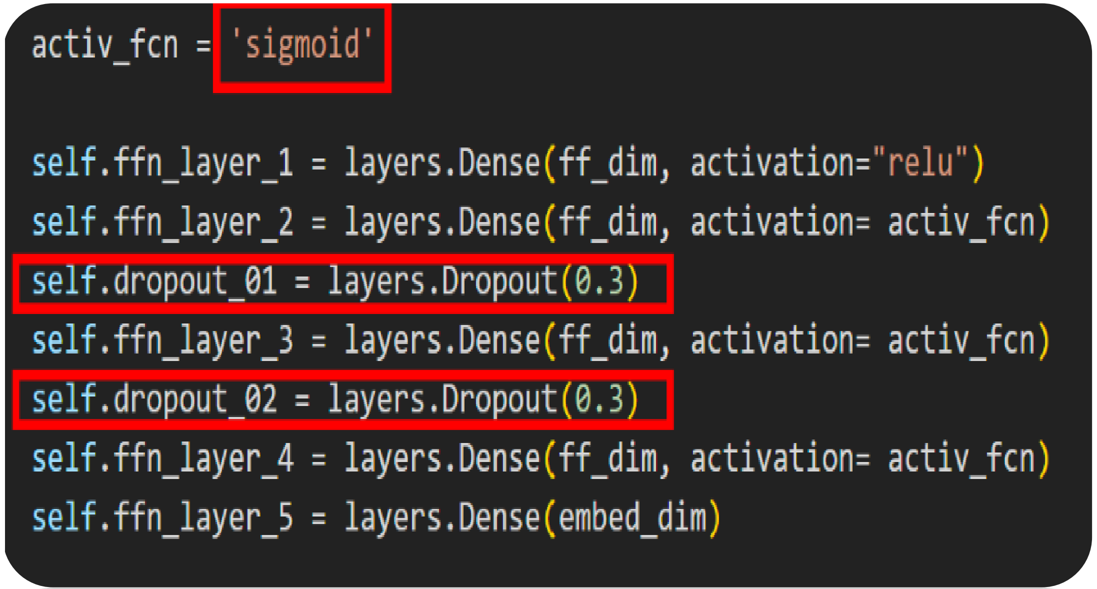
- We Trained the caption model using different CNN architectures:

CNN Model	# of epochs trained	% Train Acc	% Final Val Acc	% Train loss	% Val Loss
ResNet50	16	47.24	40.62	11.62	15.21
VGG19	16	45.46	40.15	12.12	15.31
InceptionV3	16	42.20	37.42	13.51	16.33
ResNet50V2	16	42.40	37.53	13.21	16.30
MobileNetV2	16	42.63	37.7	13.09	16.19
EfficientNetB0	17	49.10	41.00	11.02	15.03
EfficientNetB7	16	49.37	41.26	10.99	14.95

Tweaking the Text Generation Part

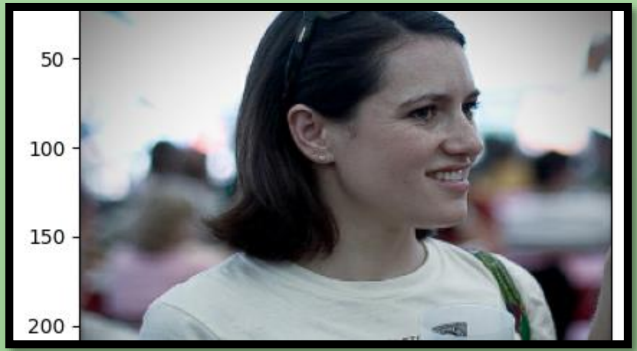


```
activ_fcn = 'swish'

self.ffn_layer_1 = layers.Dense(ff_dim, activation="relu")
self.ffn_layer_2 = layers.Dense(ff_dim, activation= activ_fcn)
self.dropout_01 = layers.Dropout(0.3)
self.ffn_layer_3 = layers.Dense(ff_dim, activation= activ_fcn)
self.dropout_02 = layers.Dropout(0.3)
self.ffn_layer_4 = layers.Dense(ff_dim, activation= activ_fcn)
self.ffn_layer_5 = layers.Dense(embed_dim)

self.layernorm_1 = layers.LayerNormalization()
self.layernorm_2 = layers.LayerNormalization()
self.layernorm_3 = layers.LayerNormalization()
```



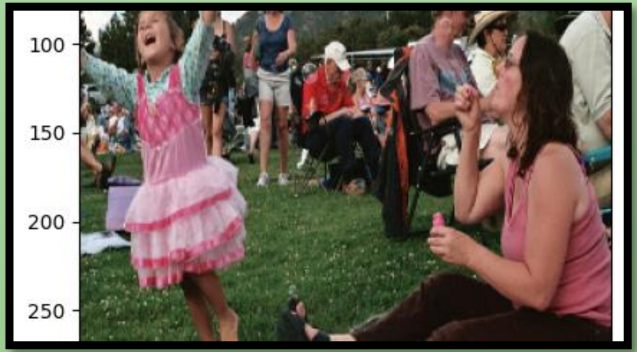
```
activ_fcn = 'sigmoid'

self.ffn_layer_1 = layers.Dense(ff_dim, activation="relu")
self.ffn_layer_2 = layers.Dense(ff_dim, activation= activ_fcn)
self.dropout_01 = layers.Dropout(0.3)
self.ffn_layer_3 = layers.Dense(ff_dim, activation= activ_fcn)
self.dropout_02 = layers.Dropout(0.3)
self.ffn_layer_4 = layers.Dense(ff_dim, activation= activ_fcn)
self.ffn_layer_5 = layers.Dense(embed_dim)
```

Tweaking the Text Generation Part



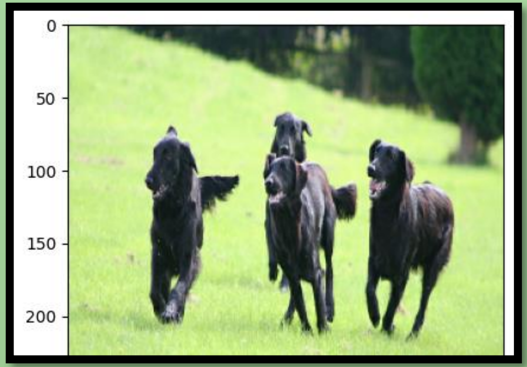
a woman with a black shirt and sunglasses on a cellphone

a woman with a white shirt and sunglasses



a group of people are standing in front of a group of people

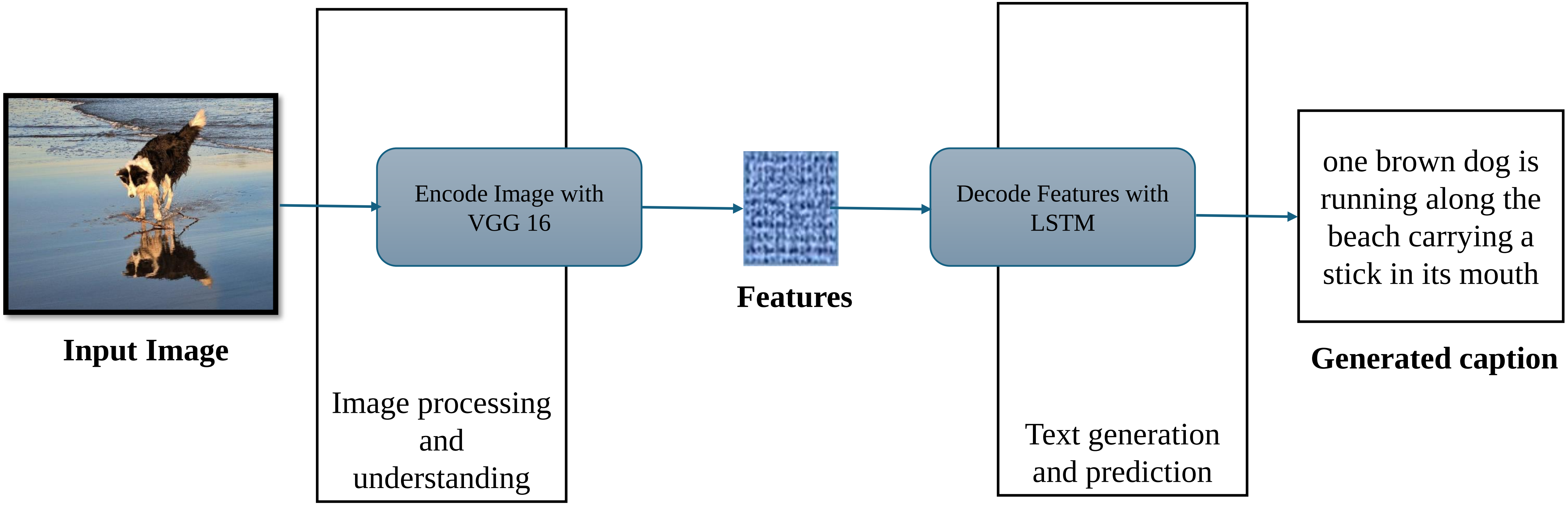
a group of people are standing in front of a group of people



two dogs run through a field

two black dogs running through a field

Caption Model 2



Encode Image with VGG 16

Decode Features with LSTM

Input Image

Features

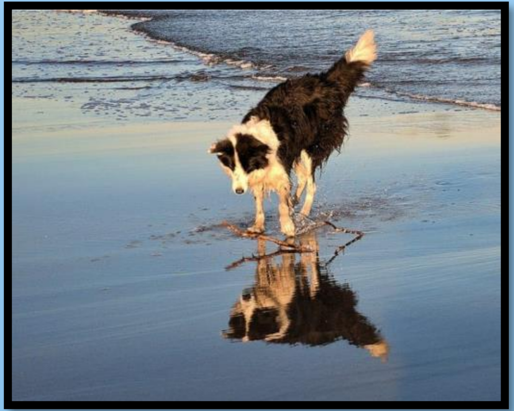
one brown dog is running along the beach carrying a stick in its mouth

Generated caption

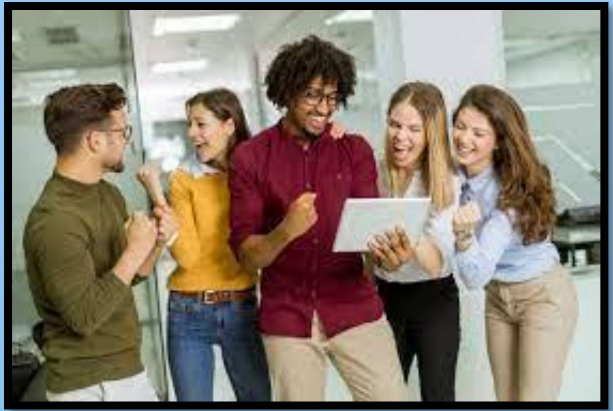
Image processing and understanding

Text generation and prediction

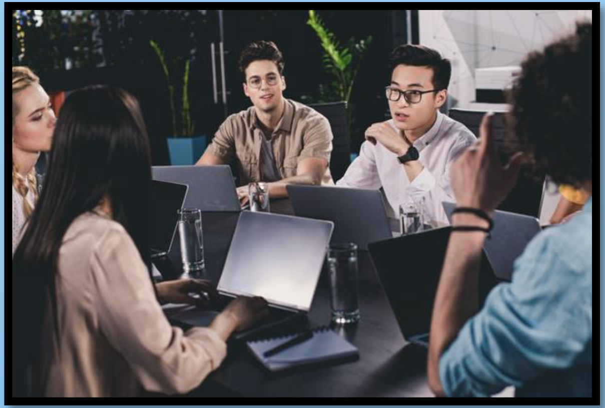
Model Performance - Results



one brown dog is running along the beach carrying a stick in its mouth



light white man is standing in front of a group of people taking pictures



taking a picture of a seated crowd of people at a bar eating a

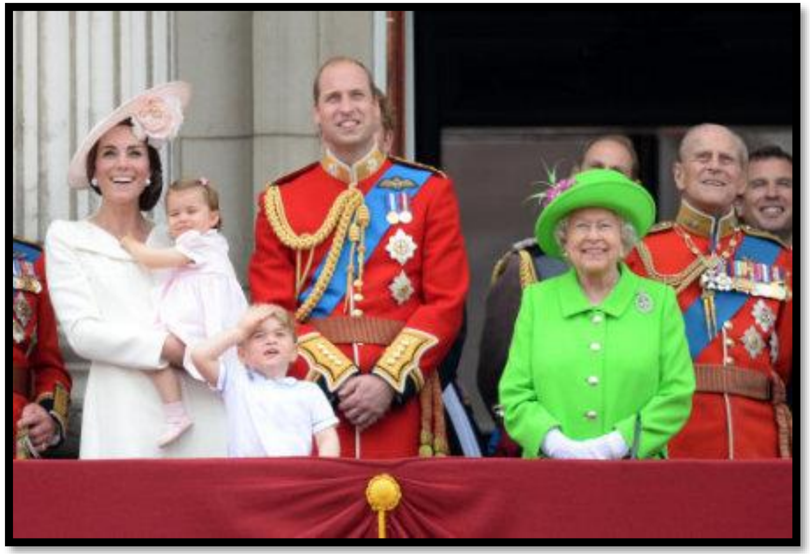
Model Performance - Limitations

- Inaccurate captions for multi-object image
- CNN may not capture all relevant features and the LSTM may lose context over longer sequences
- Generates same length of captions

of young girls are standing in front of a

of people talking on a wooden bench in front of a crowd of people

in a white shirt and white sign and white sign and a white shirt and a white shirt and



Caption Model 3

Encoder

ViT Model for Image Feature Extraction

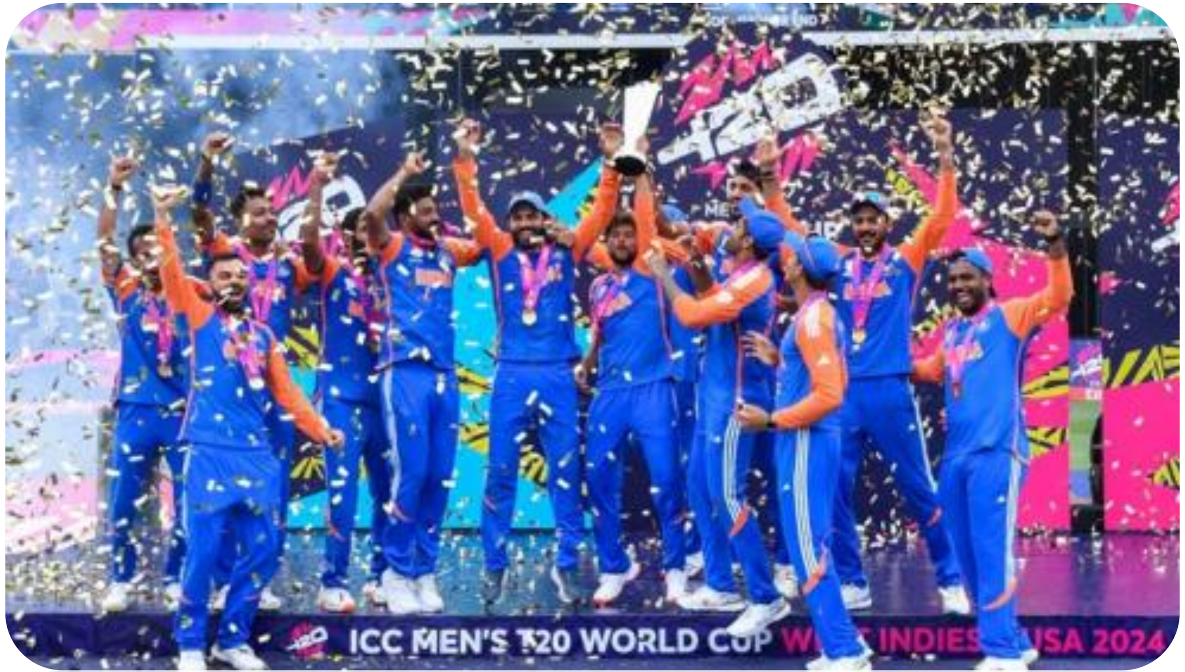
- ViT is a transformer-based
- Designed for image classification tasks.
- Splits the input image into fixed-size patches
- Flattens each patch into a vector, and then treats these vectors as input tokens.
- The output of the ViT model is a set of feature vectors

Decoder

GPT-2 Model for Text Generation

- GPT-2 is a transformer-based model designed for NLP tasks
- It takes a sequence of text tokens as input and generates a sequence of tokens as output
- Consists of multiple transformer layers with a self-attention mechanism, allowing it to capture dependencies between tokens in the input sequence.

Model Results

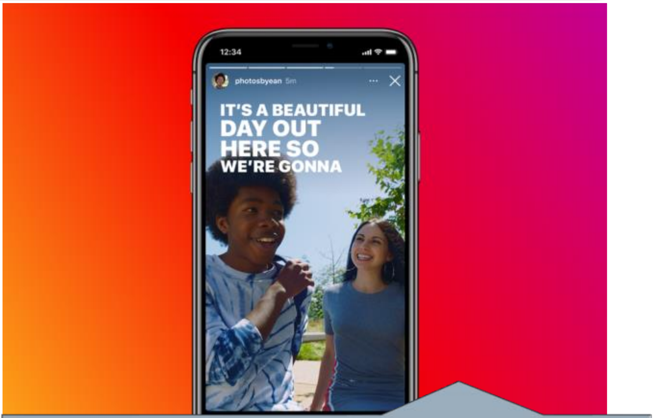


a crowd of people standing on top of a stage

Summarizing all Models

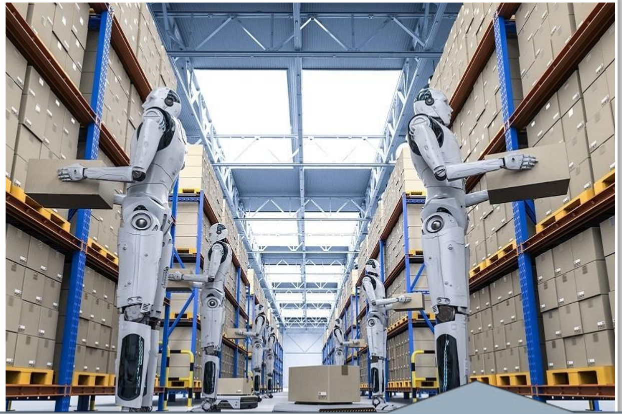
	CNN +Transformer	VGG16 + LSTM	ViT +GPT 2
Training Dataset	Flickr	Flickr	COCO
Feature Extraction	CNN	CNN	Transformer
Accuracy	High	Moderate	High
Caption relevance	High	Moderate	High
Overall Performance	Adequate	Reasonable	Superior

Real-world Applications and Deployment



Engaging Social Media Content: Generate captions that boost user interaction on platforms like Instagram and Facebook.

Enhanced Accessibility: Provide image descriptions for visually impaired individuals, fostering inclusion.

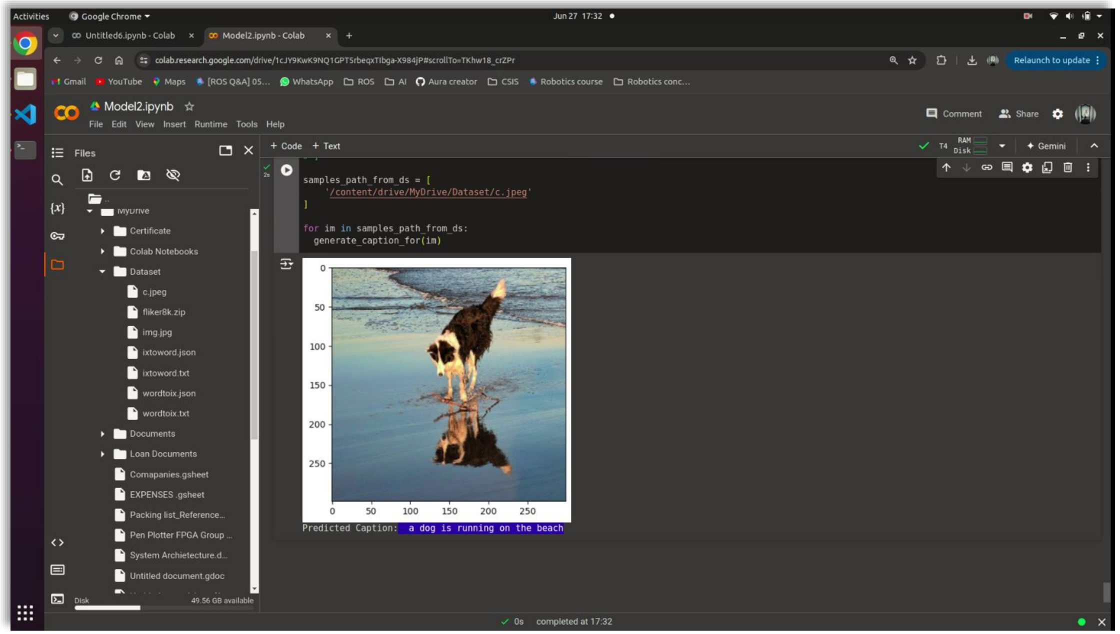


Improved Robot Navigation: Assist robots in comprehending their environment by generating captions that describe objects and actions.

Revolutionized Image Search: Enable retrieval based on content understanding, not just keywords, leading to more relevant results.

Streamlined Content Management: Automate image tagging and organization for efficient workflow.

Bridging the Gap: Deployment Considerations



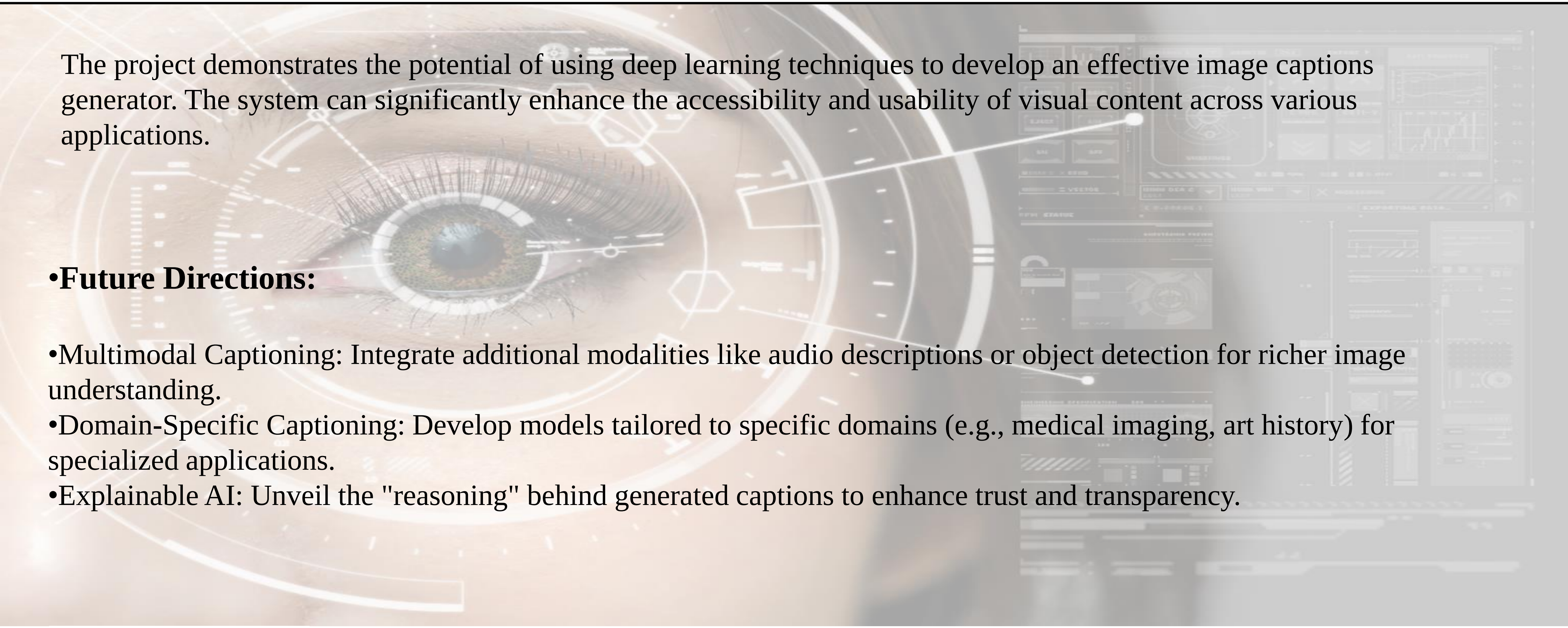
```
samples_path_from_ds = [
    '/content/drive/MyDrive/Dataset/c.jpeg'
]

for im in samples_path_from_ds:
  generate_caption_for(im)
```

```
Predicted Caption:  a dog is running on the beach
```

Hardware Requirements: The computational demands of the model will determine the hardware needed for deployment (CPUs, GPUs, TPUs).

Software Integration: The model needs to be seamlessly integrated with existing applications or platforms for user interaction.

Conclusion and Future Directions



The project demonstrates the potential of using deep learning techniques to develop an effective image captions generator. The system can significantly enhance the accessibility and usability of visual content across various applications.

- **Future Directions:**

- Multimodal Captioning: Integrate additional modalities like audio descriptions or object detection for richer image understanding.
- Domain-Specific Captioning: Develop models tailored to specific domains (e.g., medical imaging, art history) for specialized applications.
- Explainable AI: Unveil the "reasoning" behind generated captions to enhance trust and transparency.

Thank you!