SENIOR DESIGN PROJECT
END-TERM PRESENTATION

IMAGE CAPTION GENERATOR USING CNN AND LSTM

Supervised By: Dr. RANGABALLAV PRADHAN

Group No. Z10

Name of the Students with Regd. No.:

1. Nigamananda Panda Regd. No:2141016090
2. Ganesh Chandra Das Regd. No.:2141004078
3. Akashdeep Behera Regd. No.:2141002072
4. Madhab Narayan Mohanty Regd. No.:2141013300

Department of Computer Science and Engineering
Faculty of Engineering & Technology (ITER)
Siksha 'O' Anusandhan (Deemed to be) University
Bhubaneswar, Odisha

Presentation Outline

- Introduction
  - Project Overview and Problem Statement
  - Objectives and Motivation
- Background & Related Work
  - Existing Solutions/Related Work & Their Limitations/Research Gaps
- Proposed Solution & Architecture
  - System Architecture, Workflow Diagram, Model Diagram, Block Diagram, Schematic Layout
  - Description of Key Components/Features & Modules
- Implementation Details
  - Algorithms and Methods Used/Technologies & Platforms, Frameworks, and Tools Used
- Results and Analysis
  - Test Results– Performance Metrics /System Outputs and Screenshots
  - Performance Comparison/Interpretation of Results/Result Validation
- Conclusion & Future Work
  - Key Findings
  - Potential Extensions
- Bibliography

Introduction

❑ Project Overview

The Image Caption Generator with Voice is an advanced AI-based system that automatically generates descriptive captions for images and converts the captions into speech. It combines Convolutional Neural Networks (CNNs) for image feature extraction and Recurrent Neural Networks (RNNs) (specifically Long Short-Term Memory, LSTM) for generating meaningful captions. Additionally, a Text-to-Speech (TTS) module is used to convert the generated captions into voice output.

- Extract visual features from images using a **CNN model**.
- Generate meaningful captions using an **LSTM-based language model**.
- Convert text-based captions into voice using a **TTS (Text-to-Speech) engine**.
- Improve accessibility for visually impaired users by enabling voice descriptions.

Introduction

❑ Problem Statement & Motivations

With the exponential growth of digital images on the internet and social media, it has become increasingly challenging to process and interpret visual data efficiently. A major gap exists in making images more accessible to individuals who are visually impaired or those who need automated content understanding. Traditional image annotation methods require manual labeling, which is time-consuming, inconsistent, and impractical for large-scale datasets [2].

Motivations:

1. **Enhancing Accessibility for the Visually Impaired** – Providing voice-based image descriptions to improve accessibility.
2. **Automating Image Understanding and Annotation** – Reducing reliance on manual labeling with AI-driven captioning.
3. **Advancements in Deep Learning & AI** – Leveraging CNNs for feature extraction and LSTMs for natural language processing.
4. **Real-world Applications** – Useful in social media, e-commerce, and assistive technologies for content automation.

Introduction

❑ Objectives

Objectives:

- **Develop an AI-powered Image Captioning System** – Implement a deep learning model that generates accurate textual descriptions of images using CNNs and LSTMs.
- **Integrate Voice Output** – Convert generated captions into speech using Text-to-Speech (TTS) technology to enhance accessibility.
- **Improve Image Understanding & Automation** – Reduce reliance on manual annotation by automating the image captioning process.
- **Optimize Model Performance** – Train and fine-tune the model using datasets like MSCOCO and Flickr to improve accuracy and efficiency.
- **Enhance User Experience** – Ensure a seamless and user-friendly experience for visually impaired individuals and other users.

Background & Related Work

❑ Related Work & Their Limitations

1.VGG16-Based Image Captioning[4]

Description:

VGG16 is a deep convolutional neural network with 16 layers, widely used for feature extraction due to its simple and uniform architecture. It has been used in early image captioning models to extract image features, which are then passed to LSTM decoders for sentence generation.

Limitations:

- High computational cost and large number of parameters.
- Poor feature reuse; redundant filters.
- Struggles with fine-grained details due to deep, uniform layers.

2. ResNet50-Based Image Captioning[5]

Description:

ResNet50 introduces residual connections to allow very deep networks to train effectively by mitigating the vanishing gradient problem. It's been popular for image captioning because of its strong performance in feature extraction.

Limitations:

- While residual connections help, feature redundancy still exists.
- Less efficient reuse of lower-layer features compared to DenseNet.

Background & Related Work

❑Improvement of our Model

- **Dense Connectivity:** Unlike VGG16 and ResNet50, DenseNet201 connects each layer to every other layer in a feed-forward manner, allowing maximum feature reuse and preserving information from earlier layers.
- **Improved Gradient Flow:** Dense connections enable better gradient propagation during training, leading to faster convergence and higher accuracy.
- **Compact Model:** Despite its depth, DenseNet is more parameter-efficient than VGG16 or ResNet50, reducing overfitting and making it ideal for moderate-sized datasets.
- **High-Level Semantics:** Captures both fine details and abstract features, leading to more context-aware and descriptive captions.
- **Enhanced Usability with Text-to-Speech:** The addition of a voice module makes the system accessible to visually impaired users. It also provides an interactive and engaging user experience, which is not addressed in most previous works.
- **Improved Performance on Moderate Datasets:** While models like Vision Transformers require huge datasets to perform well, our DenseNet201-based system performs robustly even on limited data, due to better generalization and data efficiency.
- **Better Generalization in Diverse Scenarios:** DenseNet's capability to retain features from all layers helps in identifying objects in complex backgrounds, improving accuracy in diverse or cluttered images.

Proposed Solution & Architecture

❑ Algorithms & Methods Used

•**CNN (Convolutional Neural Networks):** Used for image feature extraction (DenseNet201). [6]

•**RNN/LSTM (Recurrent Neural Networks):** Used for sequential text generation from extracted features.

•**Attention Mechanism:** Enhances focus on relevant image regions while generating captions.

•**Text-to-Speech (TTS) Conversion:** Converts generated text into speech output.

❑ Technology Stack

- **Programming Language:** Python
- **Frameworks:** TensorFlow
- **Pre-trained CNN Models:** DenseNet201
- **Dataset:** Flickr8k
- **Text-to-Speech Engine:** gTTS
- **Development Tools:** Keras, Google Colab, Kaggle, NumPy, Pandas, Matplotlib

Proposed Solution & Architecture

- Some of the sample data taken from Flicker8k Dataset

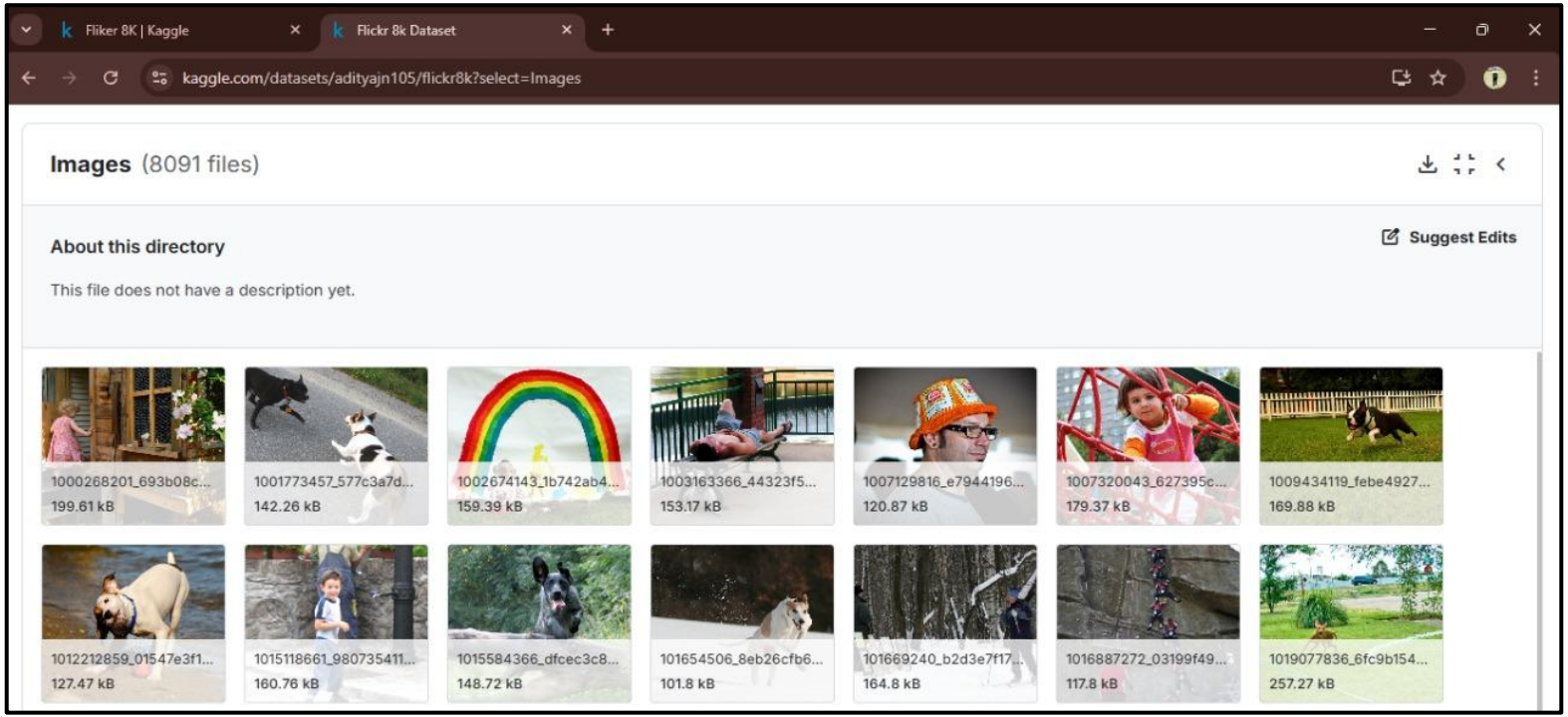
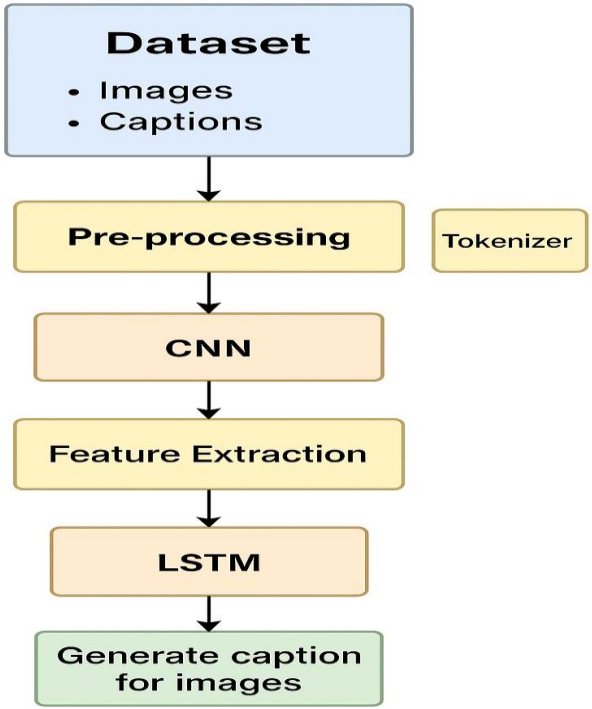


Figure 1. Snapshot of Flicker8kDataset from Kaggle

Proposed Solution & Architecture

❑Schematic Layout



Schematic Layout of Image Captioning

Figure 2. Model Diagram

Proposed Solution & Architecture

❑ Workflow

- **Image Input:** The user uploads an image.
- **Feature Extraction:** A CNN model extracts visual features.
- **Caption Generation:** An LSTM-based model processes features and generates a descriptive caption.
- **Voice Conversion:** The generated text is converted into speech.
- **Output:** The final caption is displayed and spoken aloud.

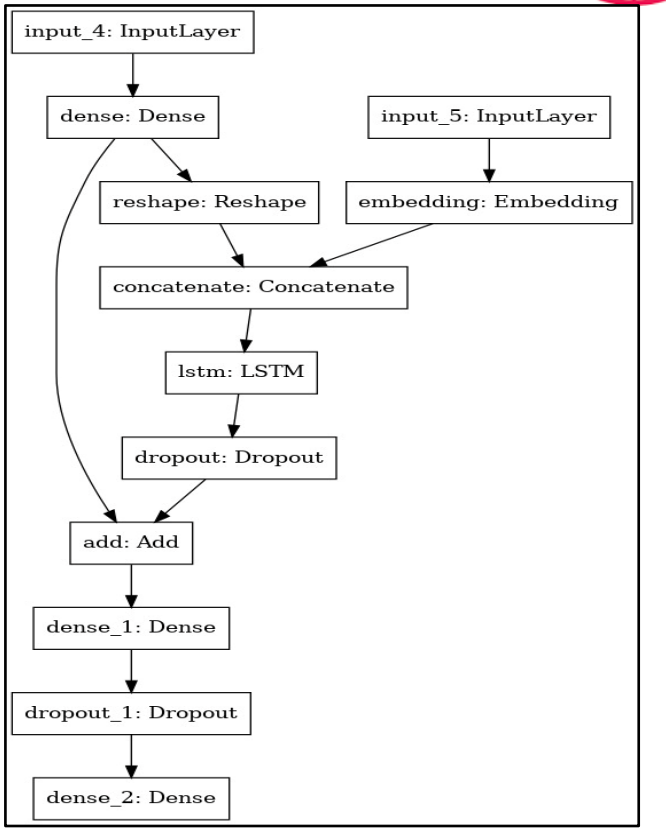


Figure 3. Model Architecture

Proposed Solution & Architecture

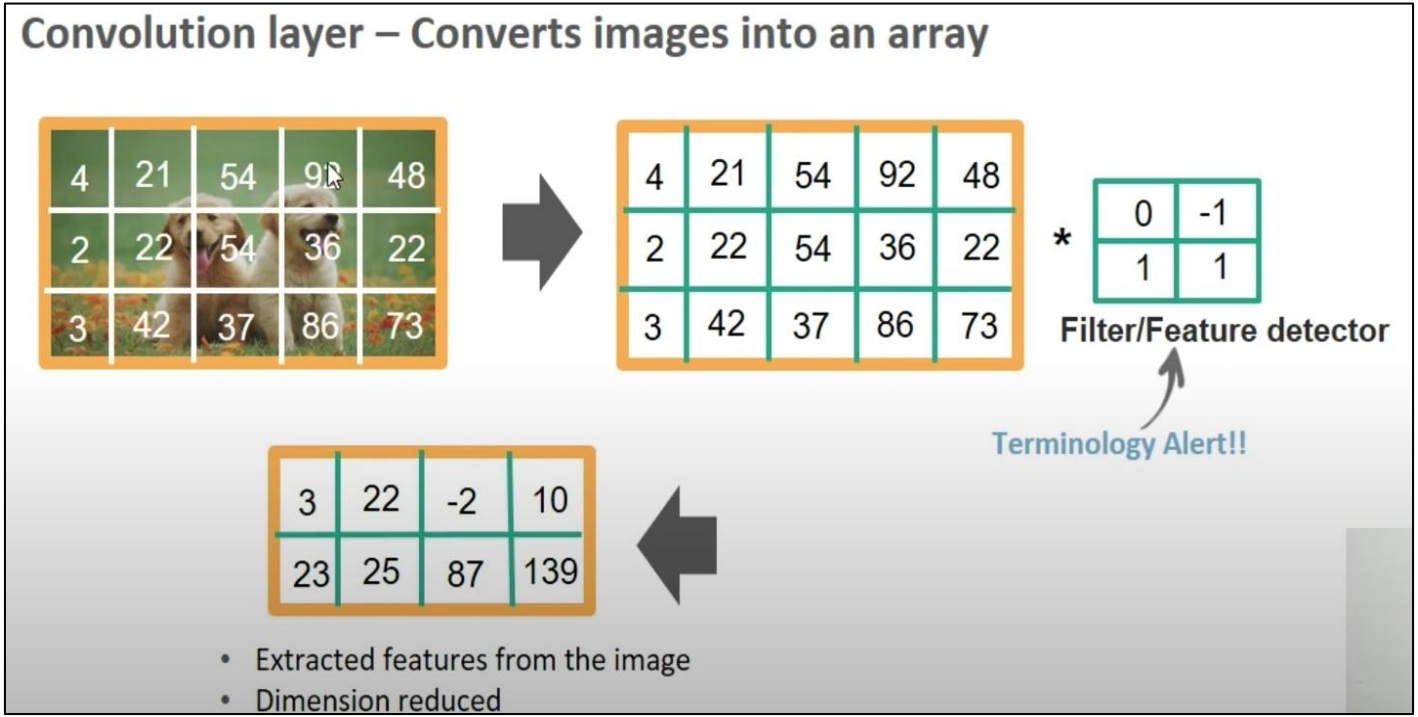


Figure 4. CNN Architecture

Proposed Solution & Architecture

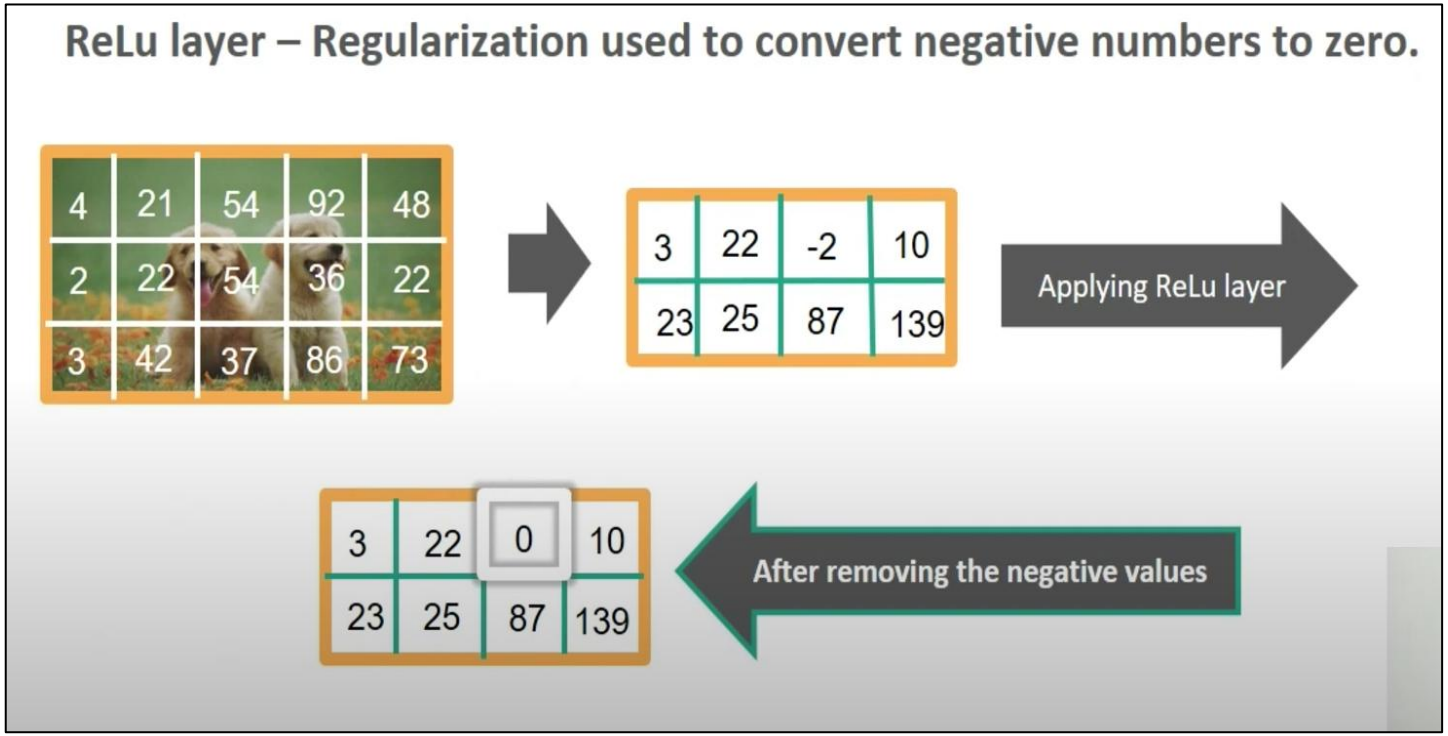


Figure 5. Relu Layer

Proposed Solution & Architecture

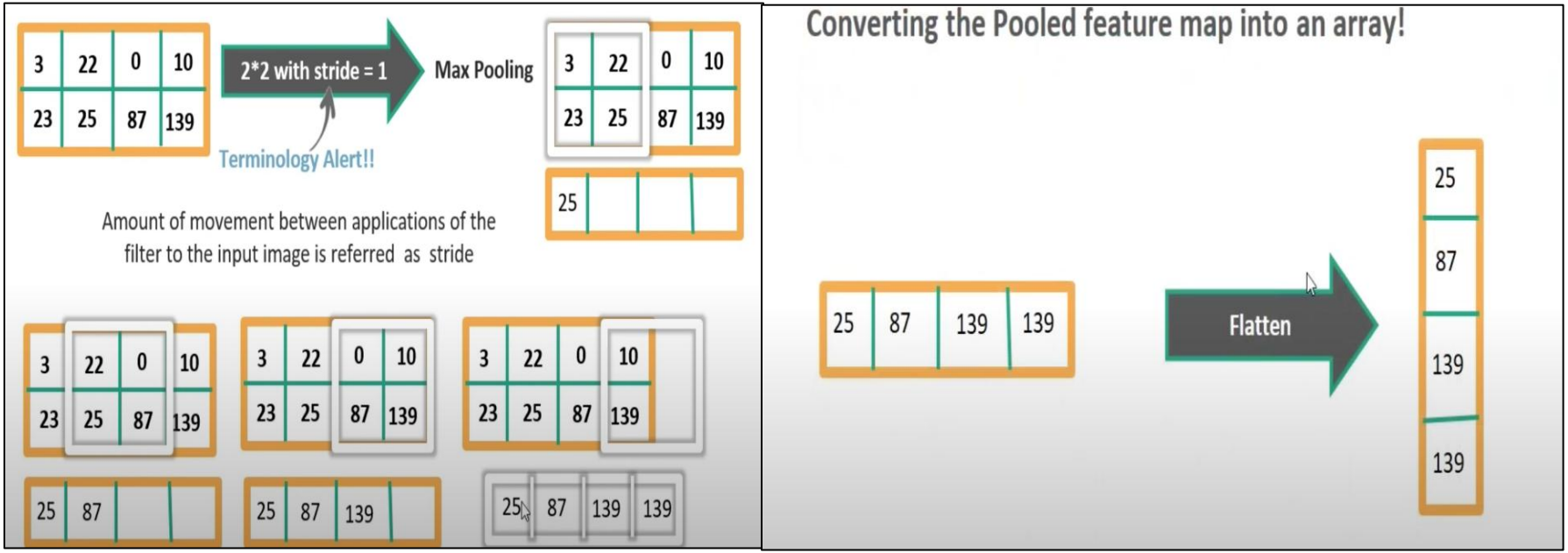


Figure 6. Pooling layer

Proposed Solution & Architecture

Fully connected layer – Combines features and produce

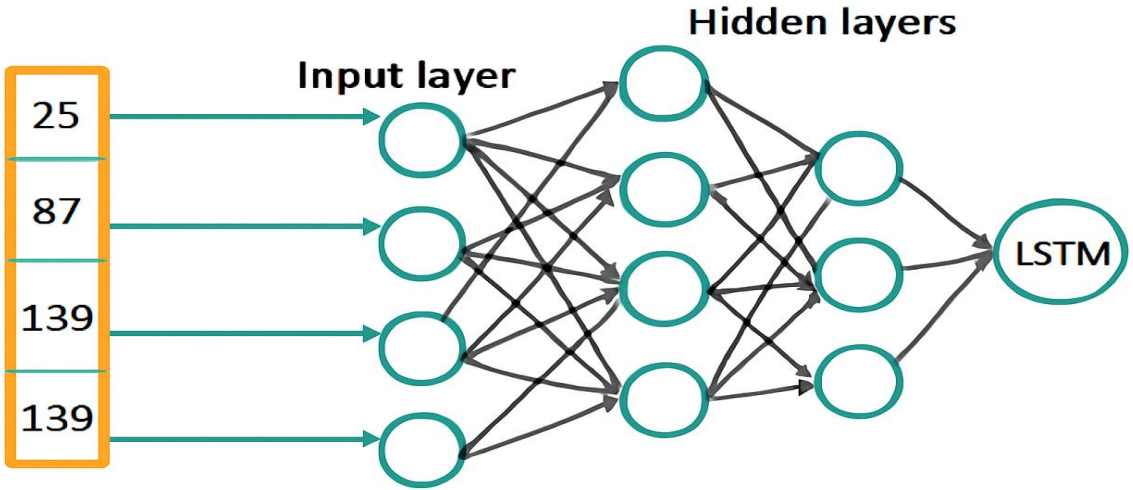


Figure 7. Fully Connected layer

Proposed Solution & Architecture

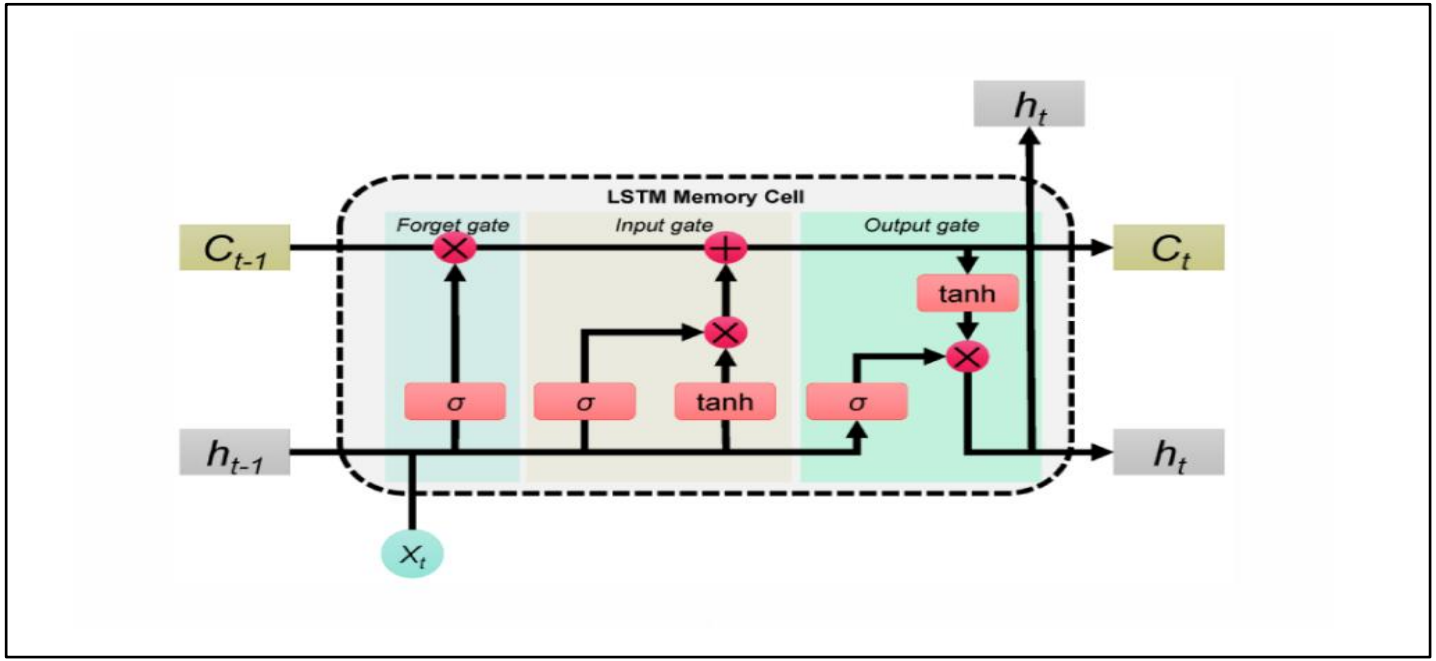


Figure 8. LSTM Architecture

Proposed Solution & Architecture

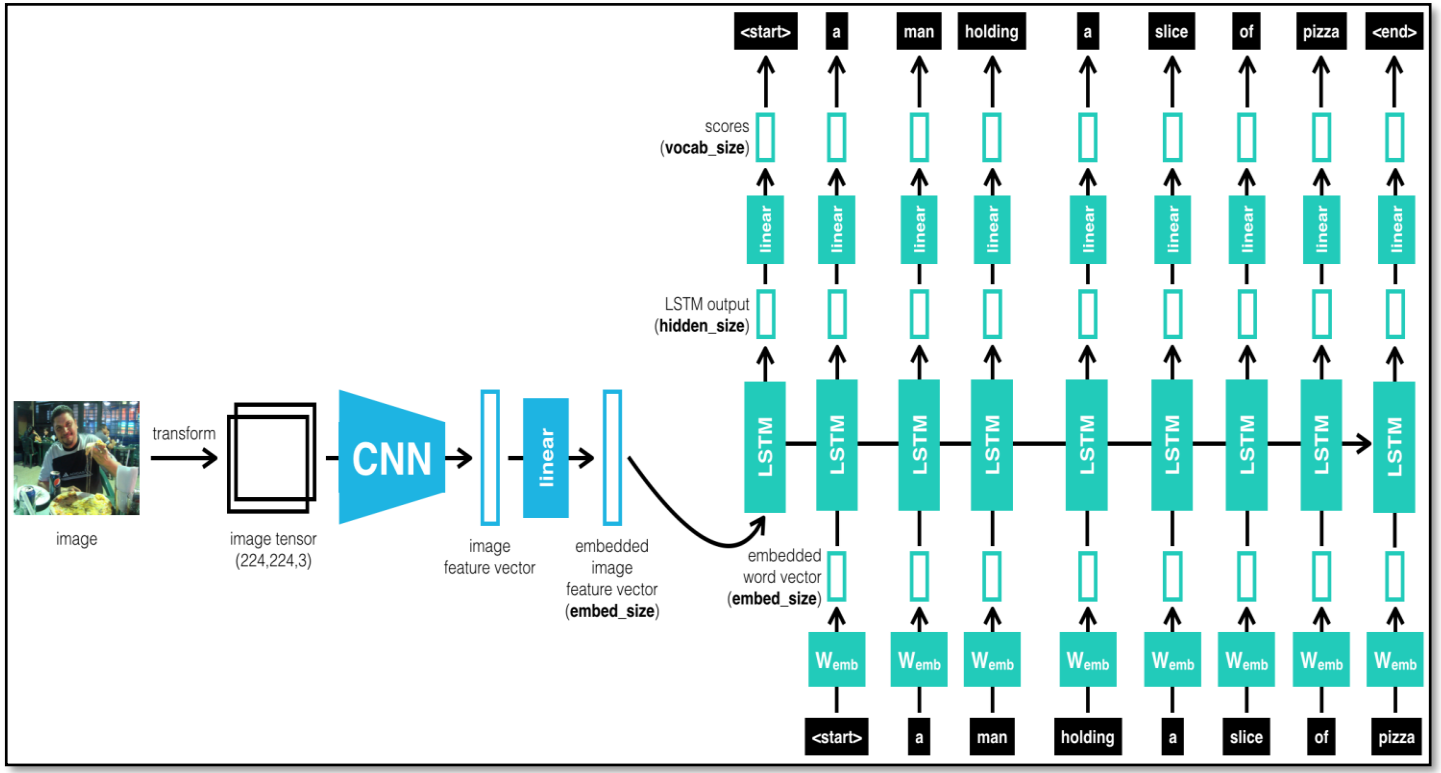


Figure 9. Concatenation of CNN and LSTM

Progress in Implementation Plan & Methodology

- **Phase 1: Research & Requirement Analysis**

- Study existing image captioning models and datasets.
- Identify key challenges and limitations in current approaches.
- Select appropriate deep learning architectures (CNN + RNN).

- **Phase 2: Dataset Collection & Preprocessing**

- Use **Flickr8k** datasets[1].
- Preprocess images (resizing, normalization) and captions (tokenization, embedding).

- **Phase 3: Model Development**

- Implement **CNN-based feature extractor** (DenseNet201)[7].
- Develop **LSTM-based caption generator** with attention mechanism.
- Train the model on preprocessed datasets.

Progress in Implementation Plan & Methodology

- **Phase 4: Testing and Optimization**

- Fine-tune hyperparameters to improve accuracy and efficiency.
- Optimize real-time processing for smooth user experience.

- **Phase 5: Integration of Text-to-Speech (TTS)**

- Convert generated text captions into speech using gTTS.
- Ensure proper synchronization of text output and audio playback.
- **Phase 6: Deployment & UI Development**
- Develop a simple **web-based or desktop UI** for image input and voice output.
- Deploy the model for real-world testing and user feedback.

Results and Analysis

❑ Performance Metrics & Outputs

❖ VGG16 Model (CNN + LSTM)

- The accuracy and loss Function of the VGG16 model are 47.6% and 1.74, respectively, as shown in the figure below.

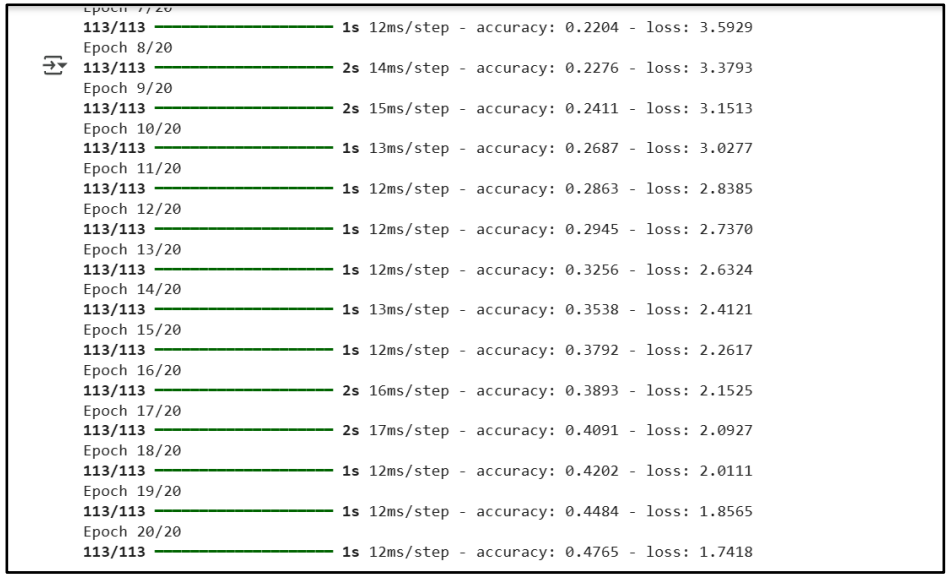


Figure 10. VGG16 Model Output Chart

Results and Analysis

❑ Graphical Representation:

❖ VGG16 Model (CNN + LSTM)

- ➢ The graphical representation of accuracy and loss for VGG16 is shown in the figure below.

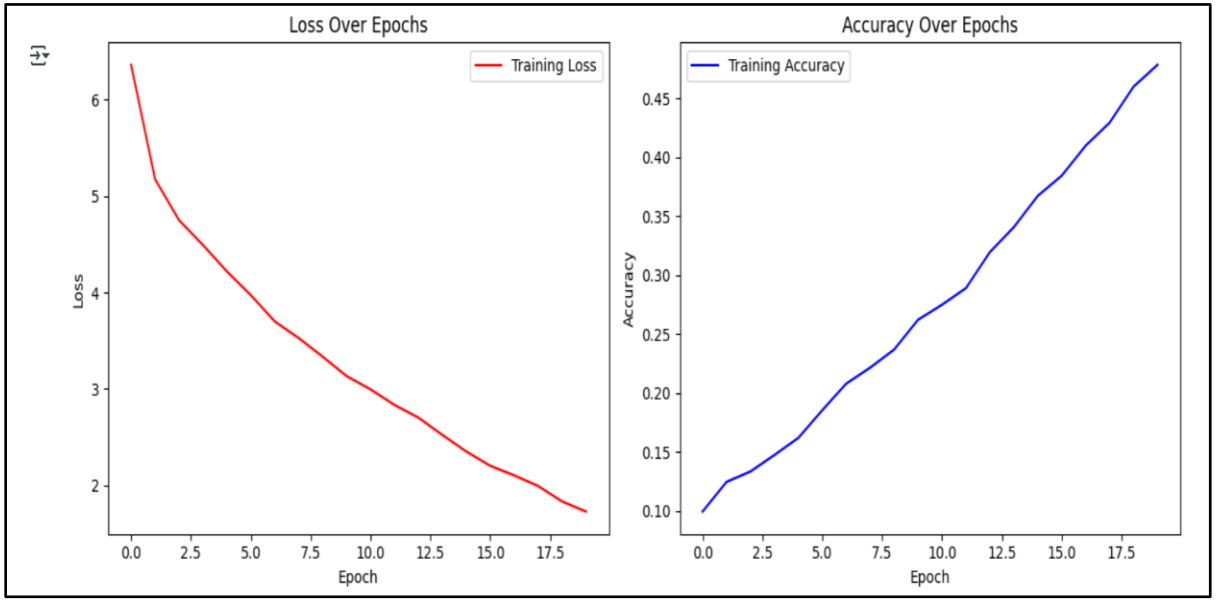


Figure 11. Accuracy and Loss Plot for VGG16

Results and Analysis

❑ Performance Metrics & Outputs

❖ ResNet50 Model (CNN + LSTM)

- The accuracy and loss Function of the ResNet50 model are 66.7% and 0.98, respectively, as shown in the figure below.

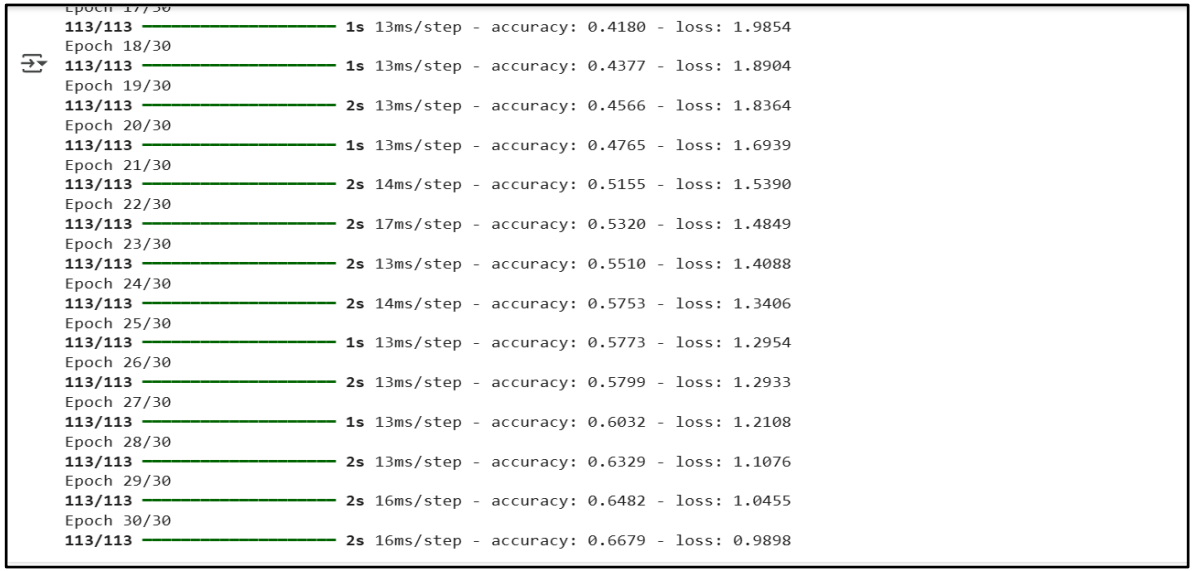


Figure 12. ResNet50 Model Output Chart

Results and Analysis

❑ Graphical Representation:

❖ ResNet50 Model (CNN + LSTM)

- ➢ The graphical representation of accuracy and loss for ResNet50 is shown in the figure below.

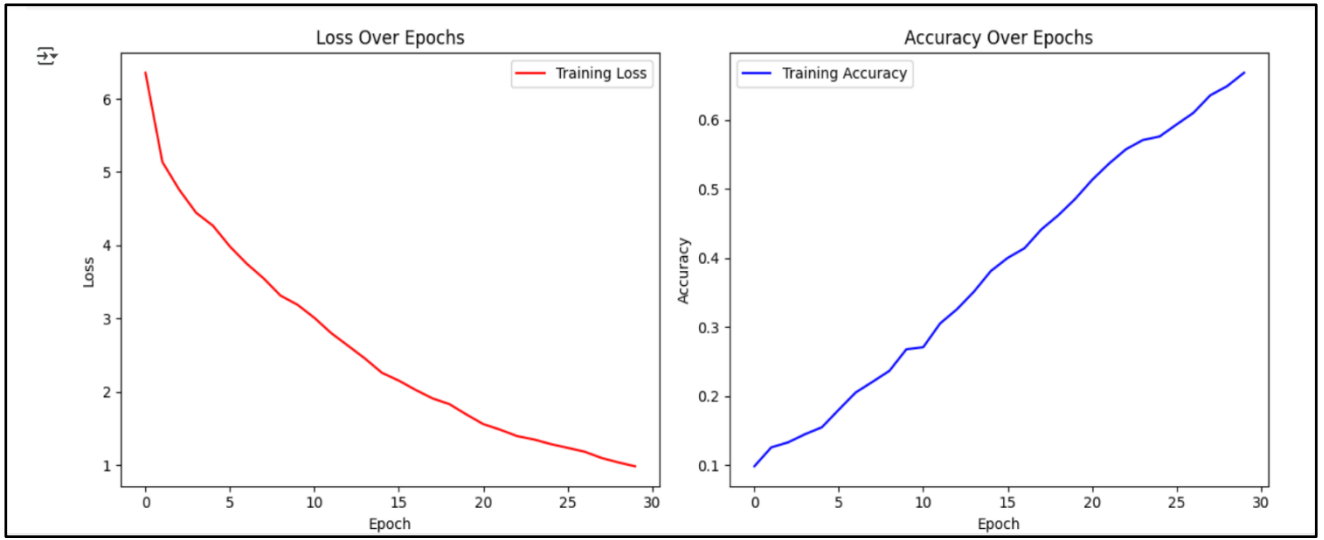


Figure 13. Accuracy and Loss Plot for ResNet50

Results and Analysis

❑ Performance Metrics & Outputs

❖ DenseNet201 Model (CNN + LSTM)

- The accuracy and loss of the DenseNet201 model are 91.2% and 0.23, respectively, as shown in the figure below.

```
Epoch 1/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 6s 3ms/step - accuracy: 0.7831 - loss: 0.6247
Epoch 2/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.8595 - loss: 0.3863
Epoch 3/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.8765 - loss: 0.3402
Epoch 4/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.8846 - loss: 0.3128
Epoch 5/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.8867 - loss: 0.3005
Epoch 6/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.8965 - loss: 0.2785
Epoch 7/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.9026 - loss: 0.2668
Epoch 8/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.9072 - loss: 0.2499
Epoch 9/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.9082 - loss: 0.2462
Epoch 10/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.9121 - loss: 0.2366
```

Figure 14. DenseNet201 Output Chart

Results and Analysis

❑ Graphical Representation:

❖ DenseNet201 Model (CNN + LSTM)

➢ The graphical representation of accuracy and loss for DenseNet201 is shown in the figure below.

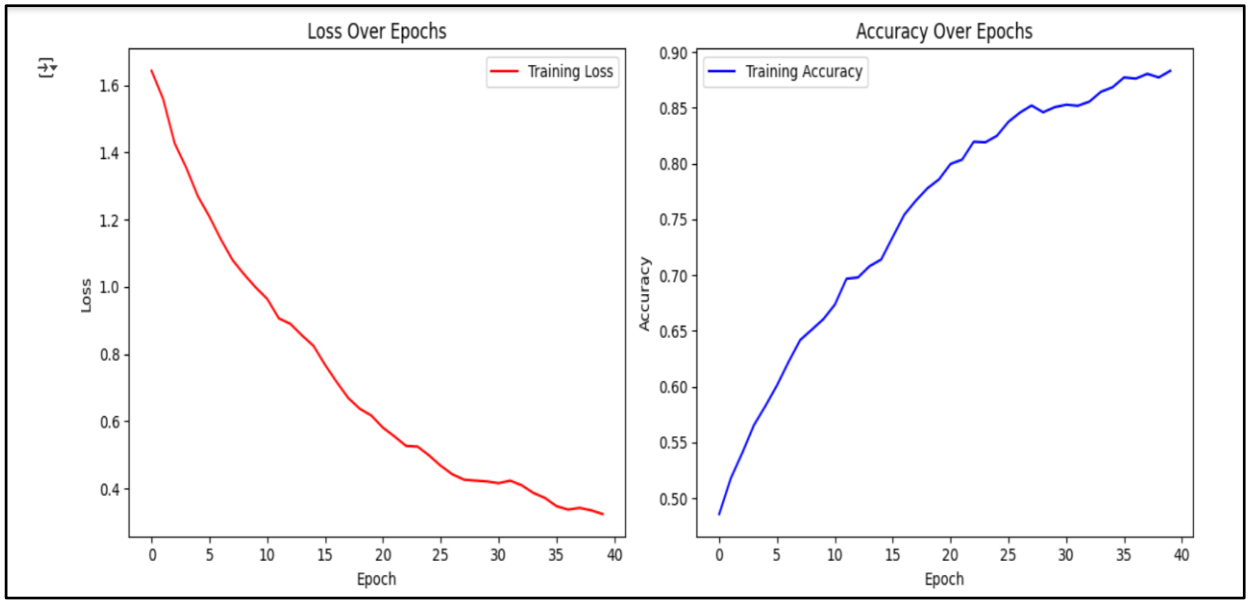


Figure 15. Accuracy and Loss Plot for DenseNet201

Results and Analysis

❑Deployment & UI Development

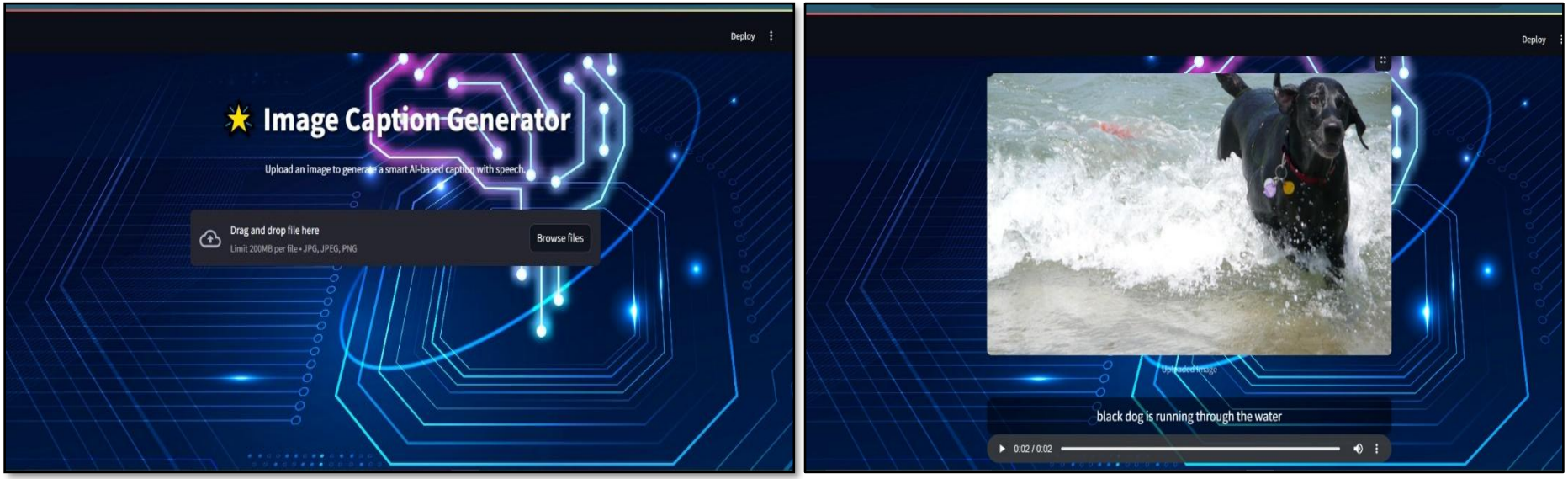


Figure 16. UI Interface

Conclusion & Future Work

❑Key Findings

- The system effectively uses CNN to extract visual features from images and LSTM to generate meaningful text captions based on those features.
- Captions are accurately formed in natural language, thanks to the sequential modeling ability of LSTM. The generated captions are then converted to speech using a text-to-speech (TTS) engine, making the system accessible for visually impaired users.
- Testing on standard datasets like Flickr8k showed good results, with captions that closely match human-like descriptions.
- The system is modular, allowing for future improvements such as attention mechanisms, real-time processing, and multi-language voice support.
- Some challenges include handling complex or cluttered images and optimizing performance for faster response.

Conclusion & Future Work

❑Potential Extensions

- **Integrate Attention Mechanism:** Improve caption accuracy by focusing on specific image regions.
- **Use Advanced CNNs:** Replace base CNN with models like EfficientNet or ResNet-152 for better feature extraction.
- **Multilingual Support:** Add multiple languages for caption generation and voice output.
- **Real-time Processing:** Optimize model for faster captioning and speech synthesis.
- **Context-Aware Captions:** Use transformers or vision-language models (e.g., CLIP) for more contextual understanding.
- **User Interaction:** Allow users to ask questions about the image via voice.

Bibliography

[1] A. J. Nigam, "Flickr8k Dataset" Kaggle,

Date of Visit: 8th June 2025, URL: https://www.kaggle.com/datasets/adityajn105/flickr8k

[2] K. Hossain, F. Sohel, M. Shiratuddin, and H. Laga, "Comprehensive survey of deep learning for image captioning" *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–36, 2019.

[3] Google Cloud, Google Text-to-Speech (gTTS) API,

Date of Visit: 8th June 2025, URL: https://cloud.google.com/text-to-speech.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, Sep. 2014.

Date of Visit: 8th June 2025 , URL: https://arxiv.org/abs/1409.1556

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[6] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks,".

Date of Visit: 8th June 2025, URL: https://arxiv.org/abs/1608.06993.


Thank
you