ProjectDesignPhase-I
Solution Architecture

Date	18November2023
TeamID	Team-592198
ProjectName	ImageCaptionGeneration
MaximumMarks	4Marks

1. **Frontend Development (HTML, CSS, JS):**

- Description: Create a user-friendly web interface for users to interact with the image caption generator. Utilize a web hosting service to deploy the HTML, CSS, and JS files.
- Integration: Ensure seamless integration with the backend for image processing and caption generation.

2. **Backend Development (Python, Flask):**

- Description: Develop a backend using Python and the Flask framework to handle user requests, process images, and generate captions.
- Integration: Connect the backend with the frontend to enable user interactions.

3. **Deep Learning Model (TensorFlow, Keras):**

- Description: Implement a deep learning model using TensorFlow and Keras for image caption generation.
- Model Architecture: Utilize a combination of Convolutional Neural Networks (CNNs) VGG16 model which contains an input layer along with the convolution, max-pooling, and finally an output layer and the LSTM model.
- Dataset: Train the model using the Flickr8k dataset for effective caption generation.

4. **Image Feature Extraction (VGG16 Model):**

- Description: Employ the VGG16 model for image feature extraction to convert input images into meaningful feature vectors.
- Preprocessing: Ensure proper preprocessing of images, converting it to lower case and remove digits, special characters and additional spaces and applying the tokenizer function to the captions before feeding them into the model.

5. **Tokenizer (Tokenization and Padding):**

- Description: Implement a tokenizer to convert text captions into numerical sequences and handle padding to ensure consistent input length for the model.

- Loading Pre-trained Components: Load pre-trained tokenizer and model components during the Flask application's startup.

6. **Model Evaluation and Building the Application**:

- Evaluation is a process during the development of the model to check whether the model is the best fit for the given problem and corresponding data. Load the saved model.

- We build our application using Flask which will run in our local browser with a user interface. In the flask application, the input parameters are taken from the HTML page These factors are then given to the model to know to predict the type of Garbage and showcased on the HTML page to notify the user. Whenever the user interacts with the UI and selects the "Image" button, the next page is opened where the user chooses the image and predicts the output.

7. **Web Application Deployment:**

- Description: Deploy the complete web application, including the frontend and backend components.

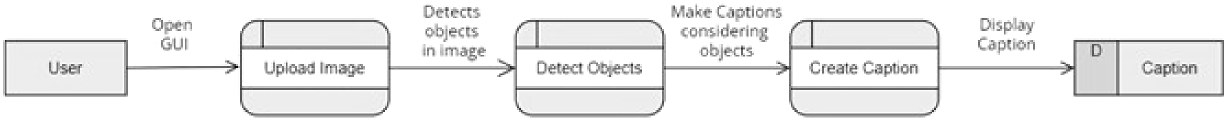

Figure1: Workflow of Image caption Generator

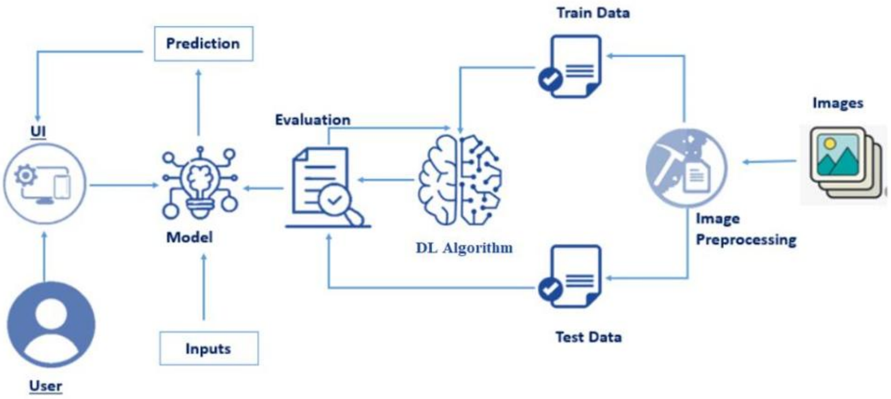

Figure2: Technical Architecture of the Image Caption Generator