FINAL PROJECT REPORT

Date	26 November 2022
Team ID	591955
Project Name	IMAGE CAPTION GENERATION

1. INTRODUCTION

1.1 Project Overview

The goal of this project is to generate the caption for image which are given input. Generally, When we see an image our brain process the information by putting the piece of what we see in the image and gives a verbal for us to tell. But how does a Machine can do? Here comes the Machine Learning algorithms and CNN concept. Which helps them to de-pixel the pics and undergo a certain process to understand what the picture stands for.

Image caption generator is a process of recognizing the context of an image and annotating it with relevant captions using deep learning and computer vision. It includes labelling an image with English keywords with the help of datasets provided during model training. These extracted features will be fed to the LSTM model, which generates the image caption.

CNN is a subfield of Deep learning and specialized deep neural networks used to recognize and classify images. It processes the data represented as 2D matrix-like images. CNN can deal with scaled, translated, and rotated imagery

In addition, the project entails integrating the created model into an accessible and user-friendly Flask web application.

1.2 Purpose

The primary goal is to design and implement an innovative system that bridges the visual-linguistic gap by generating meaningful and contextually rich captions for images. This project aims to delve into the intricacies of computer vision and natural language processing, with a focus on leveraging deep learning techniques to unravel the semantics of visual content.

In a world inundated with images, the purpose is to empower machines with the ability to not only recognize objects but also comprehend the essence of scenes. By creating an image caption generator, the project endeavours to contribute to various domains, including accessibility, where visually impaired individuals can benefit from detailed image descriptions, and content indexing, enhancing the search ability and categorization of vast image datasets.

Furthermore, the envisioned system holds potential in elevating user experiences across digital platforms. It opens avenues for interactive applications, educational tools, and immersive storytelling. Through this project, we aspire to advance the synergy between visual and linguistic understanding in artificial intelligence, unlocking new possibilities for human-machine interactions and content interpretation.

2. LITERATURE SURVEY

2.1 Existing problem

The visual world, though captivating and rich in detail, is inherently open to interpretation. Ambiguity, stemming from the subjective nature of human perception, poses a profound challenge for image caption generators. What may seem straightforward to one observer might be interpreted differently by another. This variability introduces a layer of intricacy in developing algorithms that can distill a universally agreed-upon essence from visual stimuli.

In the tapestry of visual representation, complexity often reigns supreme. Images may encapsulate intricate scenes with multiple objects, dynamic interactions, and layered contextual details. This complexity introduces a unique set of challenges for image caption generators, as they strive to distill the essence of such scenes into coherent and accurate linguistic descriptions.

One of the primary hurdles lies in deciphering the relationships between elements within a scene. Objects may interact in subtle ways, and contextual details may hold significance for the overall interpretation. Developing models that can effectively capture these nuances demands advancements in both computer vision and natural language processing.

2.2 Problem Statement Definition

The challenge at hand is extracting the image features from the image which is imported in the flask application, pre-processing the input using VGG14 and generating the caption either text form and through voice.

To create such caption generator, we will use the CNN, a subfield of deep learning and specialized deep neural networks used to recognize and classify images it processes the data represented as 2d matrix-like images, and LSTM which is type of RNN is capable of working with sequence prediction. LSTM can capture long-term dependencies in sequential data, making them suitable for generating coherent and contextually relevant captions by modelling the sequential nature of language.

3. IDEATION AND PRODUCT

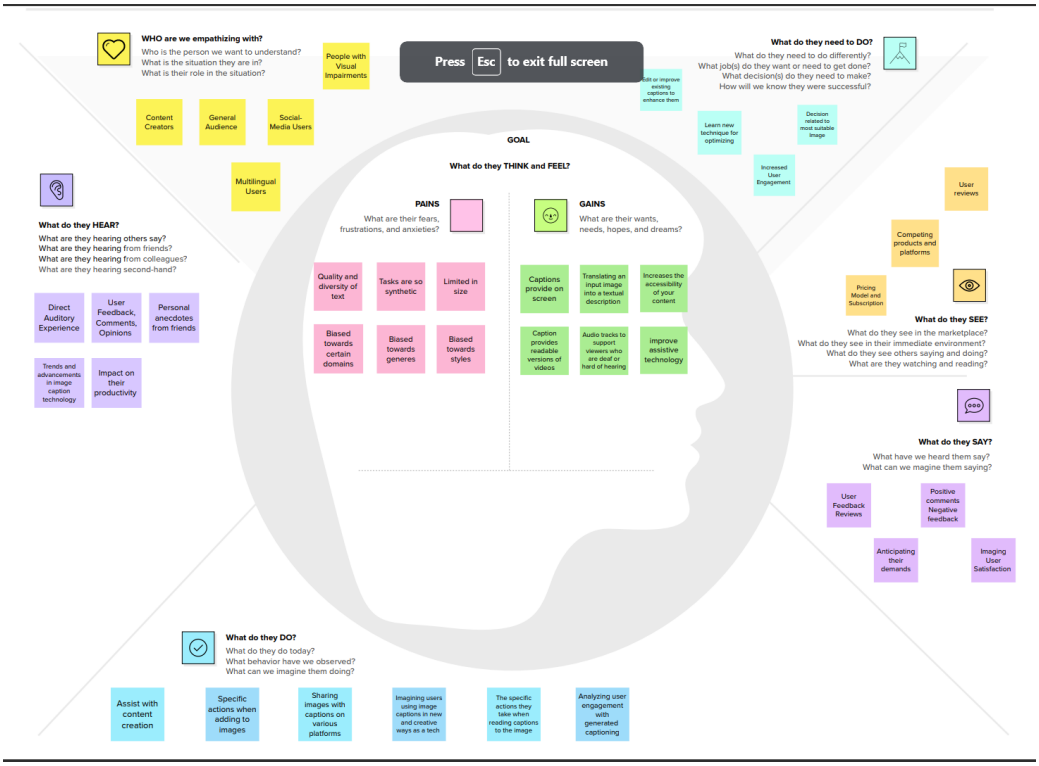


Figure – 1: Brainstorm

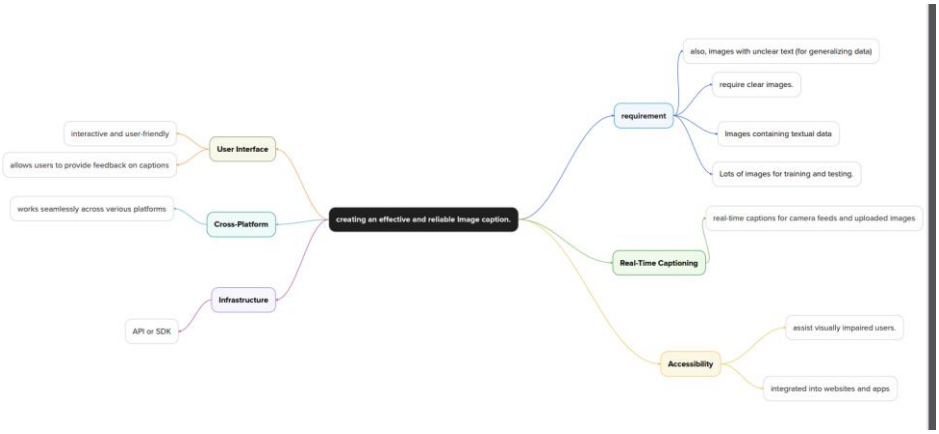


Figure – 2: Empathy mapping

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

The features and capabilities that the uploading the image and extracting its features and successfully generating the caption. The following are the requirements:

Data ingestion: here the image is ingested into the model and goes to the VGG16 for extracting the image features

Later we taring the module to improve the capacity of the model to spot of generating the exact caption for the image uploaded and testing models accuracy and performance of the trained model using a flicker dataset.

Saving the models to be later used and integrated into the flask web application, and integration of the chosen model for user-friendly web application.

4.2 Non-Functional requirement

Non-functional requirements define the characteristics and limitations to which the system must conform, guaranteeing that it satisfies the essential standards in addition to functionality.

- **User Interface Design:** An intuitive and user-friendly interface is crucial for users interacting with the image caption generator. This includes a well-designed web or application interface that simplifies the process of uploading images and accessing generated captions. A visually appealing and user-centric interface enhances the overall user experience, making the tool more accessible to a broader audience.
- **Scalability:** The ability of the system to handle a growing number of users, images, and requests without a significant decrease in performance is essential. Scalability ensures that the image caption generator remains efficient as usage scales up. Scalability is crucial for real-world applications where the system may experience varying levels of demand, ensuring consistent performance.
- **Compatibility and Integration:** Compatibility with various devices, browsers, and operating systems is important. Additionally, the ability to integrate the image caption generator with other systems or platforms, such as content management systems or social media, enhances its versatility. Compatibility and integration broaden the scope of application and facilitate seamless incorporation into existing workflows.
- **Real-Time Processing:** The ability to process and generate captions in real-time is beneficial for applications that require immediate feedback or responses. Real-time processing is essential for scenarios like live events, social media interactions, or any application where timely caption generation is crucial.
-

5. PROJECT DESIGN

5.1 Data flow Diagram

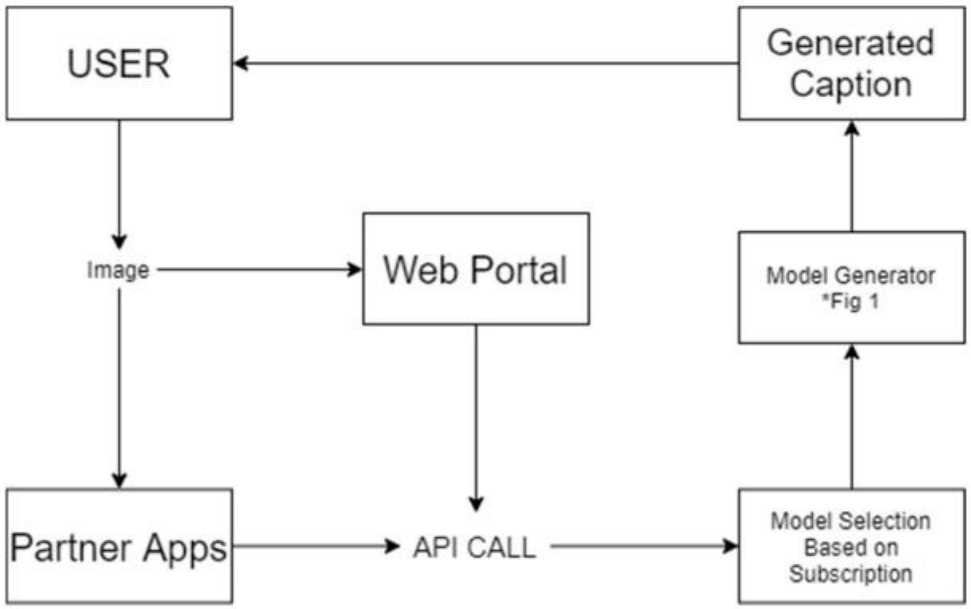


Figure – 3: Architecture

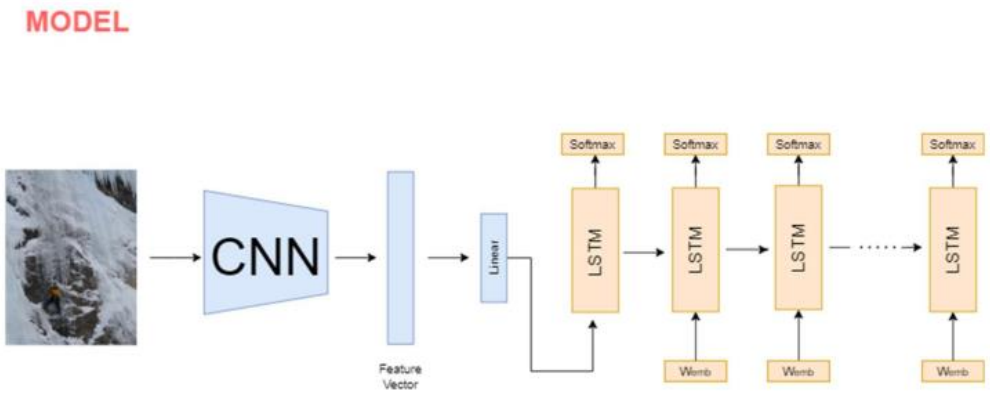


Figure – 4: CNN (Feature extracting model)

5.2 User Stories

User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook, Instagram or X	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1

Table – 1

	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
Customer (Web user)	Registration	USN-1	As a User I can register through Gamiil or any email	Receive an account activation link and get access	High	Sprint-1
		USN-2	As a User I can register through mobile number	Gets and OTP for confirmation and access the portal	High	Sprint-1
		USN-3	As a user I can register through a social media account	Confirmation of the account using the service Api	Low	Sprint-2
	Login	USN-4	As a User I can login using email, Gmail, mobile number or my social media's	Gives you access to the homepage for usage	High	Sprint-1
Subscription Model	Payment Portal	USN-1	I am able to pay using multiple options	The payment gateway works with high response and proper security measures are there.	Medium	Sprint-3
	Subscription Tier	USN-1	As a user I am able to access the free tier without any problem			Sprint-3
		USN-2	As a paid tier user, I find it very expensive	Develop more diversity in subscriptions	Low	Sprint-4
		USN-3	As a paid tier user, I expect more	Develop more feature for paid tiers	Low	Sprint-4

Table – 2

5.3 Solution Architecture

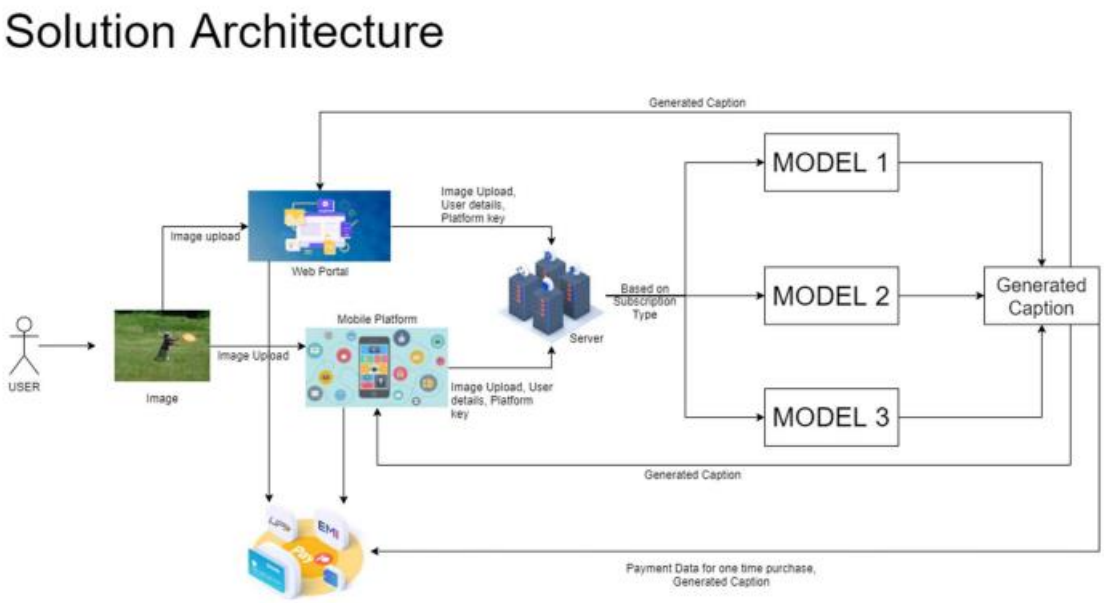


Figure – 5

6. PROJECT PLANNING

6.1 Technology Stack

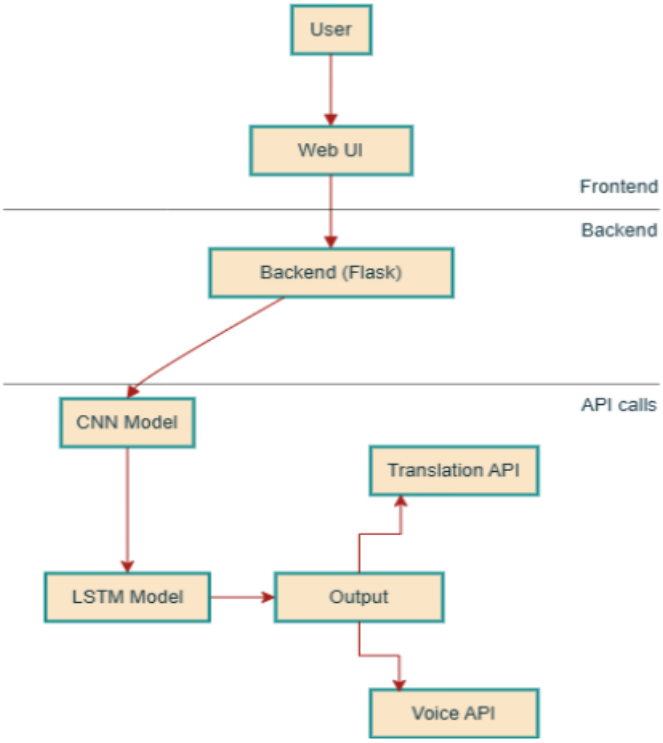


Figure – 6: Technical architecture

Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web UI with HTML, CSS, JavaScript, and Bootstrap or Tailwind for advanced CSS UI	HTML, CSS, JavaScript, Bootstrap or Tailwind
2.	Application Logic-1	Core logic for the backend process using Flask	Flask (Python Framework)
3.	Application Logic-2	Backend processing for uploaded images using CNN, features are fed into LSTM for caption generation	CNN/RNN
4.	Application Logic-3	Caption generation logic based on processed data from LSTM model	LSTM
5.	Auth Database	User Authentication and user account profile data storing in NoSQL format for use understanding and manipulation	Firebase authentication, Firebase storage (NOSQL)
6.	Cloud Database	Database for storing structured data in SQL format for rapid storage and better analysis of data	Supabase Cloud Storage (Postgre based SQL)
7.	External VOICE API-1	Third party api for direct conversion of converted image to text to voice.	RapidAPI
8.	External API-2	Integration with a Language Translation API for multi-language support and caption translation.	Google Cloud Translation, Microsoft Azure Translator, or open source api
9.	Deep Learning Model	Utilizing ML for image recognition and caption generation	Text Recognition Model, etc.
10.	Infrastructure (Server / Cloud)	Local Server Configuration: local python based environment for testing the web application. Cloud Server Configuration : Can be hosted in any suitable python based environment which can use Deep learning model in backend to process image and send output to client system.	Local System, Platforms like: Python Anywhere, Render, Heroku, Google Cloud, etc.

Table – 3

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilization of open-source frameworks, such as Flask (Python Framework) for backend and Bootstrap/Tailwind for advanced CSS UI in the front end.	Flask (Python Framework), Bootstrap, Tailwind, HTML, CSS, JS
2.	Security Implementations	Implementation of security measures including the use of AES-256-GCM for encryption for data protection, Single signin methods (only one user can sign in at a time) and Gmail authentication	Single login system, Gmail auth, AES-256-GCM encryption
3.	Scalable Architecture	Adoption of a scalable architecture, utilizing microservices for modular development and can be upgraded from flask backend to Django for robust performance and management.	Flask can be extended to Django application, changing infrastructure from monolithic to microservices.
4.	Availability	Ensuring high availability through the use of load balancers for efficient distribution of incoming traffic, distributed servers for redundancy, and other measures to prevent service downtime.	Load Balancers, Distributed Servers, Redundancy Measures, HTTPS/TLS.
5.	Performance	Design considerations to optimize performance, including managing the number of requests per second, implementing caching mechanisms for quicker data retrieval, and leveraging Content Delivery Networks (CDNs) for efficient content distribution.	Caching, CDNs, Gunicorn for optimizing performance

Table – 4

6.2 Project planning

Sprint	SPRINT NAMES	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Project Setup and Initial Features (Ideation and brainstorm)	Project visions and stakeholders	USN-1	As a user, the need to upload an image, so that it can generate caption for it	2	High	Full Team
Sprint-2	Caption Generation, (Project Desing Phase)	Image Upload feature and basic image recognition model integration	USN	User: the ability of the system to generate concise and relevant captions for uploaded image .	1	Medium	Full Team
Sprint-3	Project Planning and Technology phase	Technology used and time to lay the plan of the					
Sprint-4	Model Building, coding, Model Training	Designing model and Implementing model training capabilities and continuously improve accuracy	Task	Task: Integrate mechanisms for data ingestion and preprocessing during the training and ser up a continuous integration pipeline for automatic model update	2	High	Full Team
Sprint-5	Documentation and	Document the project and prepare for deployment and finalizing	Task	Documenting the project and including user guides and prepare for deployment	1	High	Full Team

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	4 Days	23 Oct 2023	26 Oct 2023	18	27 Oct 2023
Sprint-2	20	5 Days	28 Oct 2022	1 Nov 2023	20	2 Nov 2023
Sprint-3	20	5 Days	02 Nov 2023	6 Nov 2023	17	14 Nov 2023
Sprint-4	20	10 Days	9 Nov 2022	18 Nov 2023		
Sprint-5	20	4 Days	18 Nov 2023	21 Nov 2023		

Table – 5

7. CODING & SOLUTIONING

7.1 Feature 1: Image Caption Generation using CNN/RNN and LSTM

The image caption generation feature represents a pivotal component in our project, leveraging cutting-edge deep learning techniques. Primarily, the process involves feature extraction using a pre-trained VGG16 model to obtain rich image representations. These features are then fed into an LSTM model responsible for generating contextually relevant captions. The model is trained using categorical cross entropy as the loss function and the Adam optimizer for optimal convergence.

7.2 Feature 2: Novelty - Voice Generated Captions for Visually Impaired

In a stride towards inclusivity, our project introduces a novel feature aimed at enhancing accessibility for visually impaired users. This feature enables the conversion of generated captions into voice using a Text-to-Speech (TTS) library. The gTTS library, for instance, allows us to seamlessly convert textual captions into audible content, providing an alternative means for users to perceive image descriptions.

8. PERFORMANCE TESTING

8.1 Performance Metrics

In the realm of performance testing, our focus spans across critical metrics to guarantee the robustness and efficiency of our system. The Caption Generation Time is meticulously measured to assess the speed at which captions are generated for a given set of images. Accuracy, a paramount metric, is evaluated through BLEU scores and human assessments, ensuring that our captions align with user expectations. Resource utilization, monitored during peak usage, safeguards against potential bottlenecks by keeping a keen eye on CPU and memory consumption.

In our pursuit of ensuring optimal system functionality, we employ comprehensive metrics that delve into both speed and accuracy.

Metrics:

- **Caption Generation Time:** Measured meticulously to gauge the speed of caption generation for a set of images.
- **Accuracy:** Evaluated through BLEU scores, a standard metric for assessing the quality of generated captions. BLEU-1 and BLEU-2 scores provide insights into unigram and bigram precision, respectively.
- **Resource Utilization:** Monitored during peak usage, safeguarding against potential bottlenecks by keeping a keen eye on CPU and memory consumption.

Results:

- **BLEU-1 Score:** 0.543390
- **BLEU-2 Score:** 0.314341

These BLEU scores serve as a quantitative representation of the quality and precision of our generated captions, reflecting our commitment to delivering contextually relevant and accurate image descriptions.

9. RESULTS

9.1 Model Screenshots

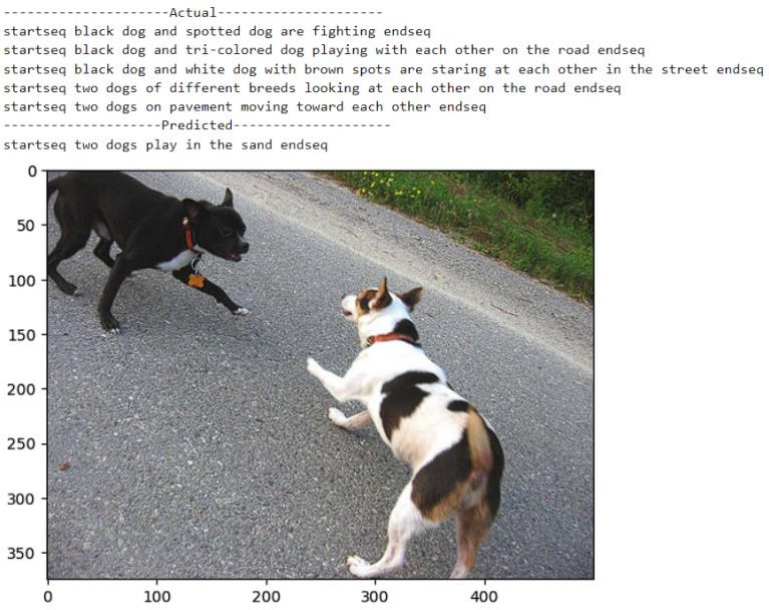


Figure – 7: Validation - 1

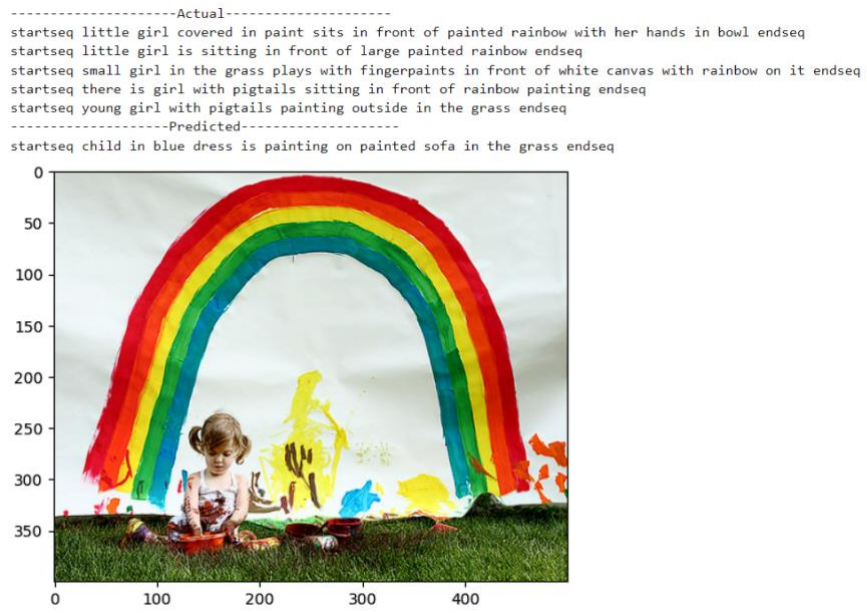
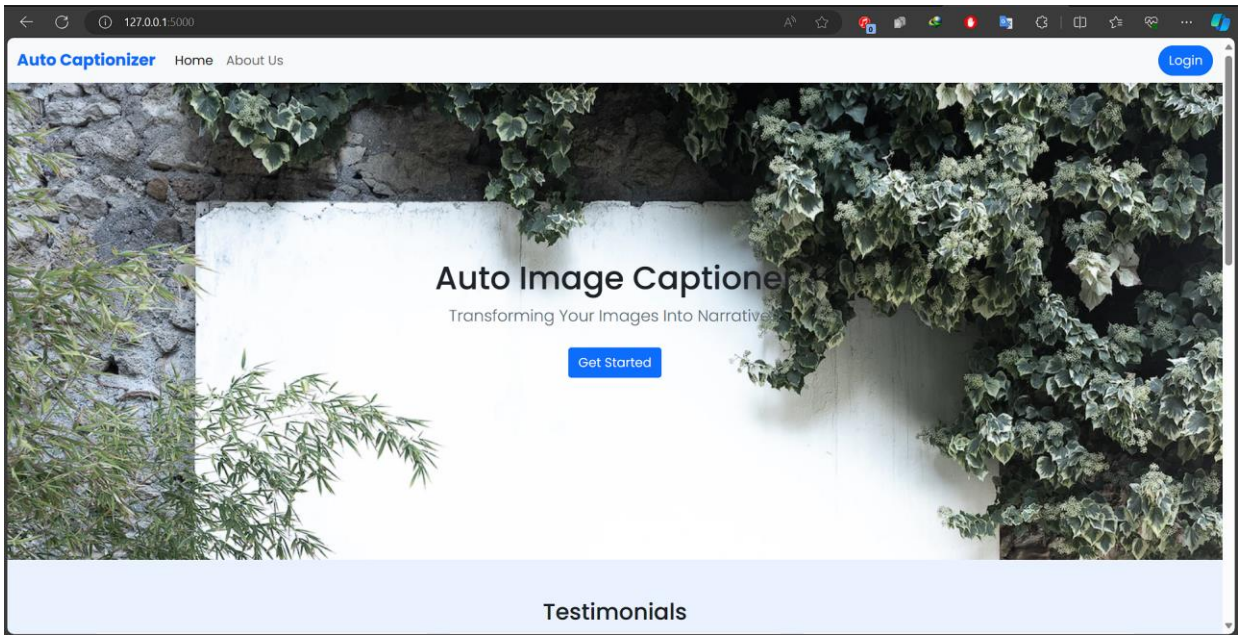
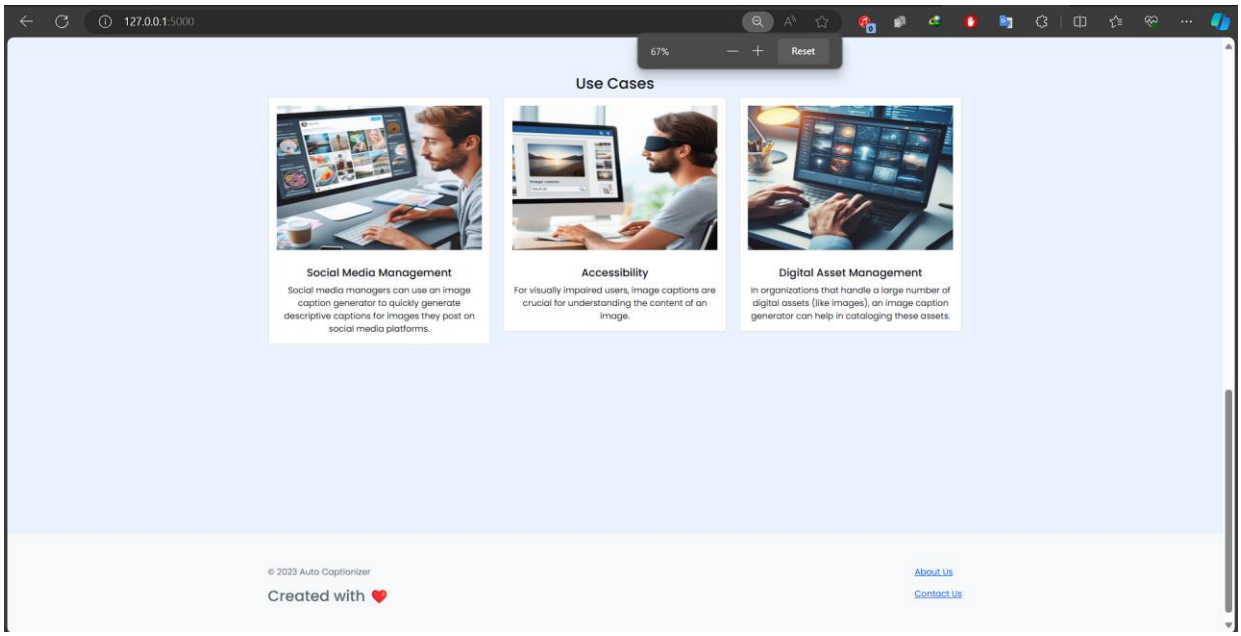


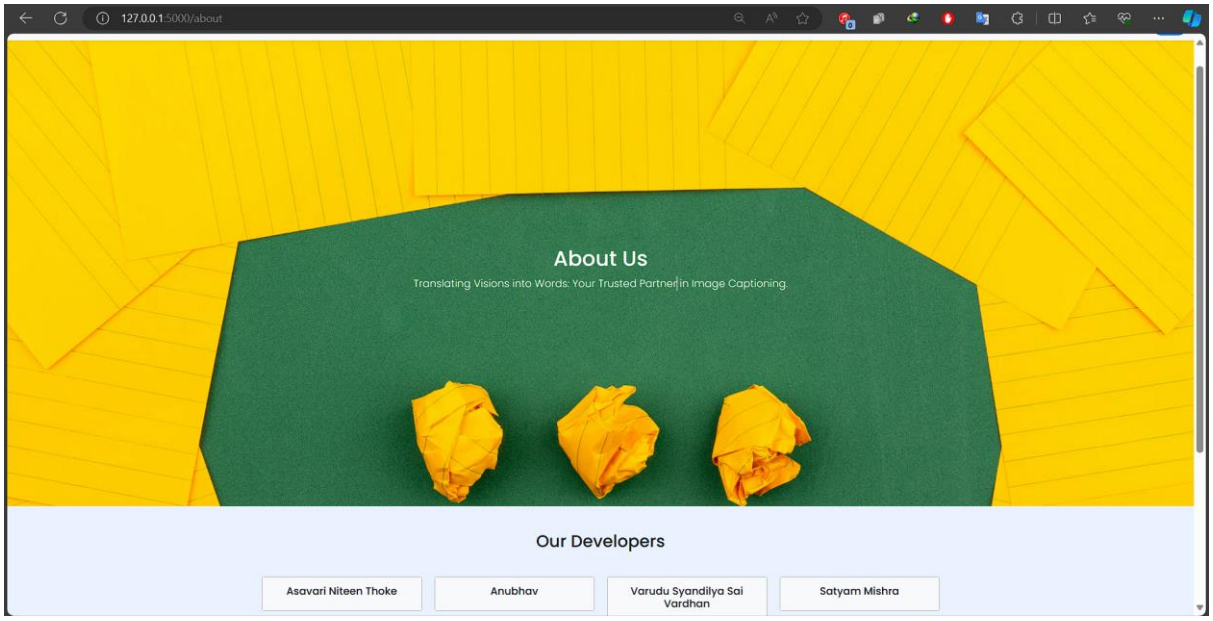
Figure – 8: Validation - 2

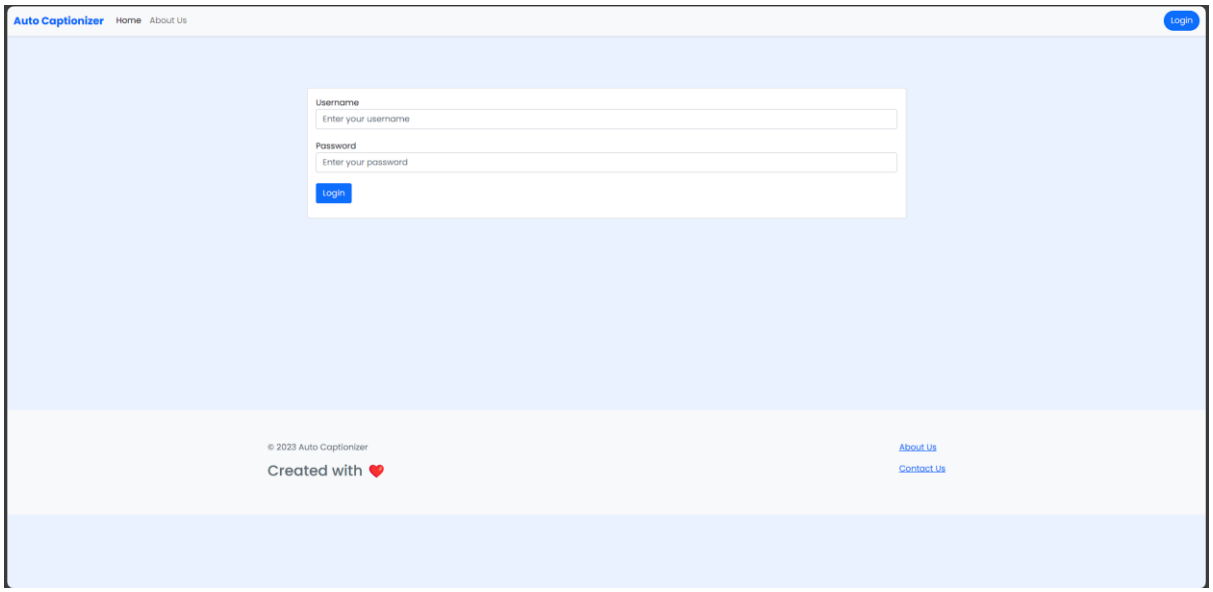
As visual representations speak louder than words, our output screenshots exemplify the fruition of our project. Figure 1 and 2 showcases the Image Caption Generation examples, where users witness the generation of diverse and contextually relevant captions.

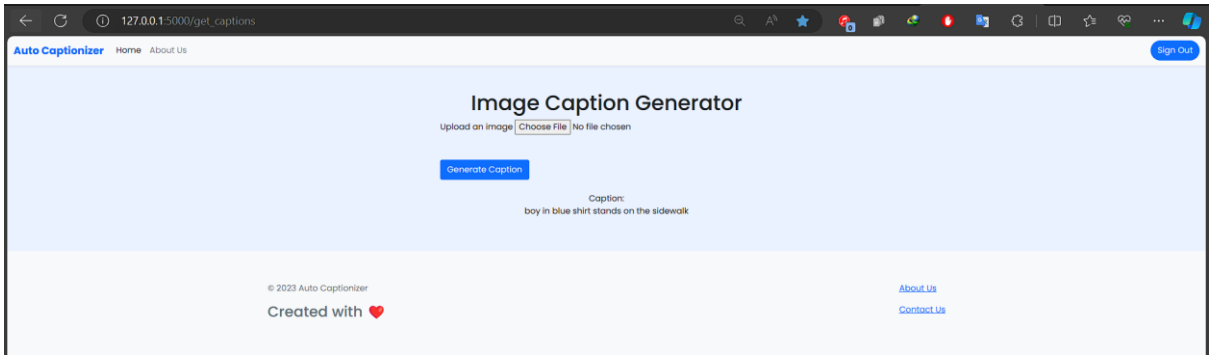
9.1 Web Interface Screenshots

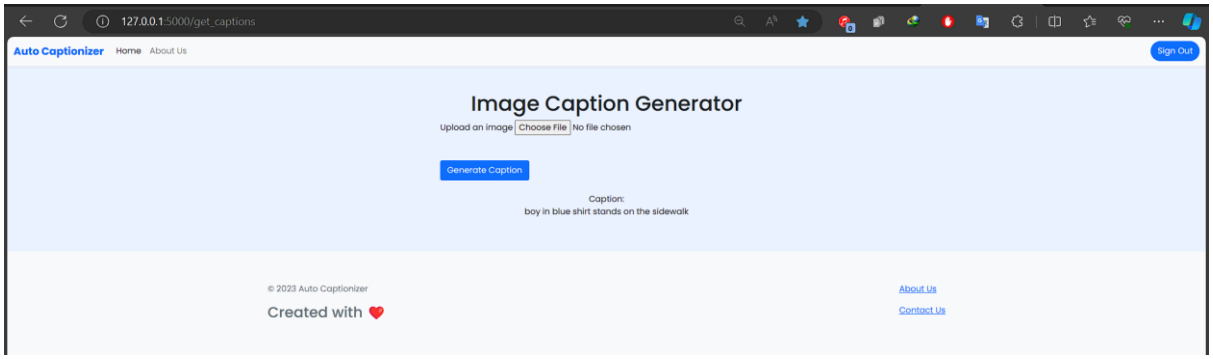





127.0.0.1:5000/about
About Us
Translating Visions into Words: Your Trusted Partner in Image Captioning.
Our Developers
Asavari Niteen Thoke
Anubhav
Varudu Syandilya Sai Vardhan
Satyam Mishra


Auto Captionizer
Home
About Us
Login
Username
Enter your username
Password
Enter your password
Login
© 2023 Auto Captionizer
Created with
About Us
Contact Us





All Webapp interface images here

Nestled within our project's architecture is a Flask web interface that serves as the visual gateway to our intricate backend machine learning model. This web service, adorned with a sleek and user-friendly design, invites users to embark on an interactive journey with our cutting-edge caption generation system. The interface showcases an array of features, prominently displaying the ability for users to effortlessly upload images.

10. ADVANTAGES & DISADVANTAGES

Advantages:

Our project brings forth several advantages, notably diverse and contextually relevant captions that enhance user engagement. The inclusion of voice-generated captions fosters inclusivity, ensuring a more accessible experience. Additionally, potential revenue streams are envisioned through subscription models and batch processing capabilities.

Disadvantages:

Despite the advantages, our system's efficacy is contingent on the quality and diversity of the training dataset. Inaccuracies in generated captions may arise, emphasizing the need for ongoing refinement and enhancement.

11. CONCLUSION

In summary, our project marks a significant stride in the realm of image caption generation, introducing a Flask-powered web interface that orchestrates a harmonious interaction with our advanced backend machine learning model. The culmination of Convolutional Neural Networks (CNN) for feature extraction and Long Short-Term Memory (LSTM) networks for caption generation is seamlessly encapsulated within the user-friendly embrace of the Flask web service. The interface not only demystifies the complexities of machine learning but also presents users with an elegant gallery, showcasing uploaded images alongside their contextually relevant captions. This visual representation serves as a testament to the project's success in bridging the gap between cutting-edge technology and user accessibility.

12. FUTURE SCOPE

The future trajectory of our project includes continuous model improvement with larger and more diverse datasets, ensuring unparalleled accuracy. Integration with popular social media platforms is on the horizon, enabling seamless caption generation directly within users' preferred platforms. Further exploration of advanced captioning techniques is anticipated to refine our system's capabilities.

13. APPENDIX

Source Code:

The source code for our project is available on our GitHub Repository.