Generating Captions from Images and Vice Versa

Hamza Safdar

21075860

UWE Bristol

Abstract

Image Captioning is a challenging task in the field of computer vision, involving the generation of natural language descriptions for images. This task requires understanding the visual content of an image, as well as the ability to generate grammatically correct and semantically meaningful sentences. In this project, I propose a new image captioning model that utilizes a combination of convolutional neural networks (CNNs) and long short-term memory networks (LSTMs) to generate captions. I used 3 different CNNs (VGG16, DenseNet201, and ResNet50). Then some modifications to the already proposed CNN+LSTM architecture which includes adding image feature embeddings to the output of the LSTMs and then passed on to the fully connected (FC) layers. Our model is trained on a Flicker 8K dataset of images and captions and can generate accurate and descriptive captions for a wide range of images. We evaluated the performance of our model using BLUE and METEOR scores and demonstrated its superiority over existing models.

Motivation

The motivation behind this project is to build an application that helps users to enhance their experience by using artificial intelligence. In the following ways this application can benefit its users:

1) Editing Applications
2) Assistance for the Visually Impaired
3) Social Media
4) Image Understanding
5) Human-computer interaction

Problem Statement

This problem introduces an artificial intelligence task, which uses computer vision (CV), deep learning (DL), and natural language processing (NLP) techniques to automatically understand the contents of images and describe them in words, on the other hand, it converts the words or sentence into a visualization or image. Users can translate the captions into different languages.

Input:
Text/Image

Output:
Text/Image

Task:
Mapping Between Text and Images

Dataset

Name:

Flickr 8k Dataset

Source:

Kaggle

Images:

8091 Instances

Data Preparation

Image Features Extraction:

- VGG16
- DenseNet201
- ResNet50

Caption Processing:

- Cleaning
- Tokenization

VGG16

The VGG16 model is a deep convolutional neural network architecture that is known for its high accuracy in image classification tasks.

It is characterized by its use of multiple convolutional layers with small filters (3x3), and it uses a large number of parameters and a deep architecture. Due to its depth, it is prone to overfitting, which can be addressed by adding regularization techniques.

The VGG16 returns us the 4096 features for each image.

DenseNet201

It is an extension of the DenseNet architecture, which connects all layers in a feed-forward fashion, where each layer receives the feature maps of all preceding layers as inputs.

DenseNet201 is known for its ability to maintain high accuracy while having a lower number of parameters compared to other architectures, which is beneficial for computational efficiency and reducing the risk of overfitting.

The DenseNet201 returns us the 1920 features for each image.

ResNet50

It is an architecture based on a **Residual Network (ResNet)** which introduces a new concept called identity shortcut connection which allows the model to avoid the vanishing gradient problem by adding the input to the output of the layer.

ResNet architectures are known for their ability to **handle very deep networks** and still maintain **good performance**.

The ResNet50 returns us the **2048 features** for each image.

Comparison

In summary, VGG16 is known for its depth, DenseNet201 for its ability to maintain high accuracy while having a lower number of parameters, and ResNet50 for its ability to handle very deep networks and still maintain good performance.

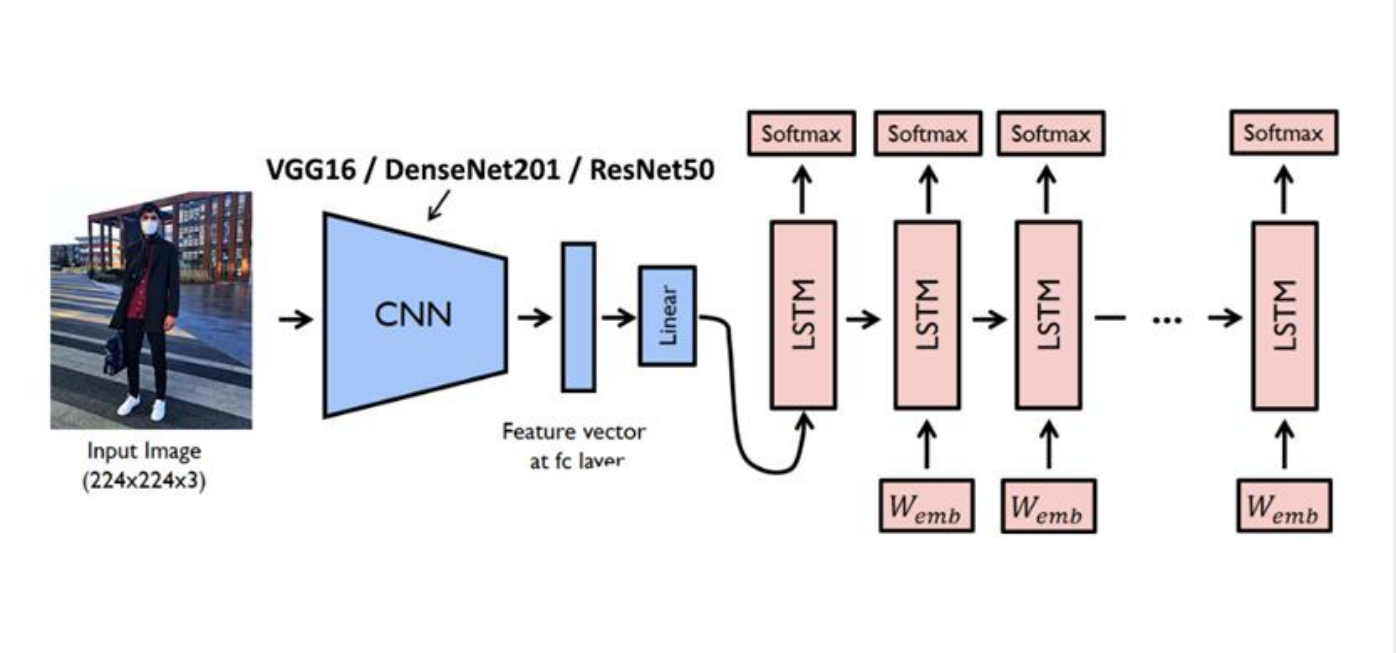
Text Cleaning

To pre-process the Captions, I have used the following techniques:

- Lowercasing:
- Removing Punctuation and Special Characters:
- Replacing Extra Spaces:
- Adding Start and End Tages:
- Tokenization:

Data Split		
Sample Data	100 %	8091 Instances
Training Data	90 %	7281 Instances
Testing Data	10 %	810 Instances

Model Architecture



Tunning

Two increase the performance of the model I did a few modifications to the proposed model architecture that was already described.

- I have added image feature embeddings to the output of LSTMs. Then passed this to the fully connected layers.
- It improved the model's performance.

Evaluation Measures

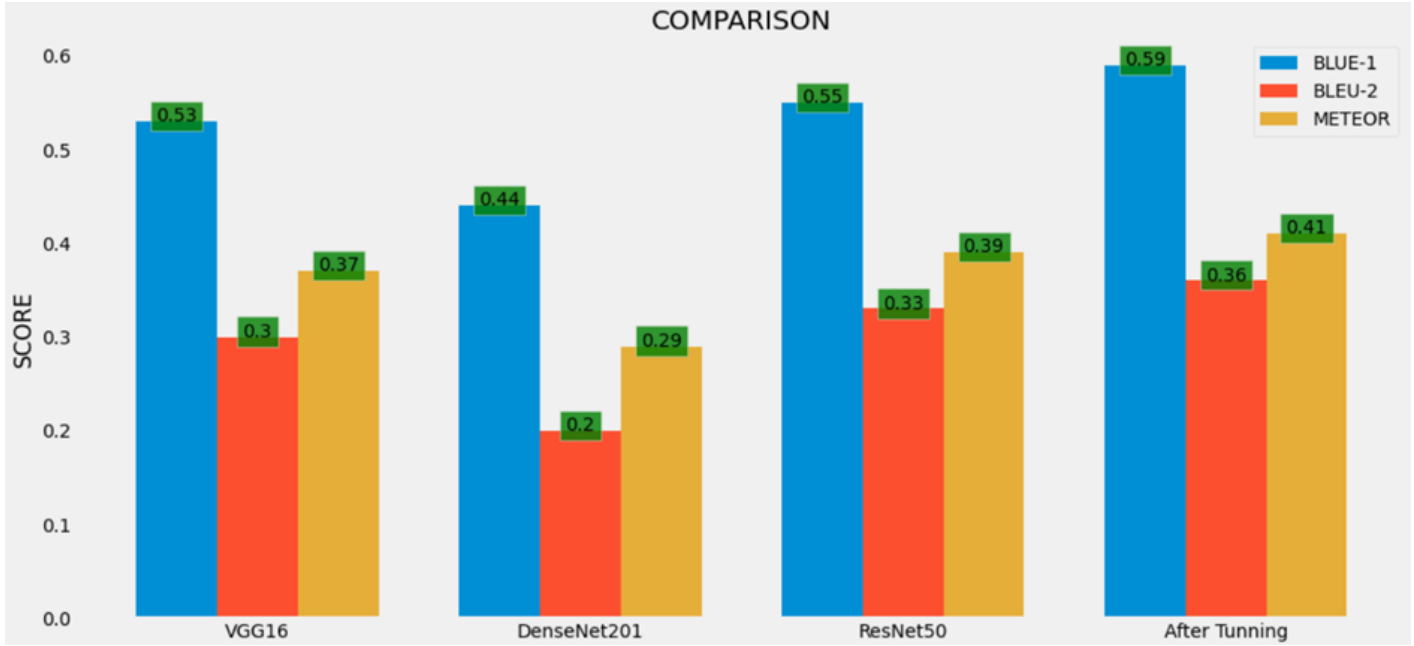
BLEU Score:

- BLEU (Bilingual Evaluation Understudy) is a method for evaluating the quality of machine translation. It works by comparing the machine-generated translation with one or more reference translations, and it calculates a score between 0 and 1, with 1 being a perfect match.

METEOR Score:

- METEOR (Metric for Evaluation of Translation with Explicit Ordering) is a method for evaluating the quality of machine translation. It is similar to BLEU in that it compares the machine-generated translation with one or more reference translations, but it also takes into account the synonymy and word alignment between the machine-generated translation and the reference translation.

Performance Evaluation



```
================================================================================
                                Actual Captions
================================================================================

man in hat is displaying pictures next to skier in blue hat
man skis past another man displaying paintings in the snow
person wearing skis looking at framed pictures set up in the snow
skier looks at framed pictures in the snow next to trees
man on skis looking at artwork for sale in the snow

================================================================================
                               Predicted Captions
================================================================================

two people are standing on the edge of mountain dome

================================================================================
                                Selected Image
================================================================================
```

Results

- Results

Enter Caption:A man is wearing white jacket

Deploying Model through FLASK

FLASK Framework

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

Interfaces
Web Application

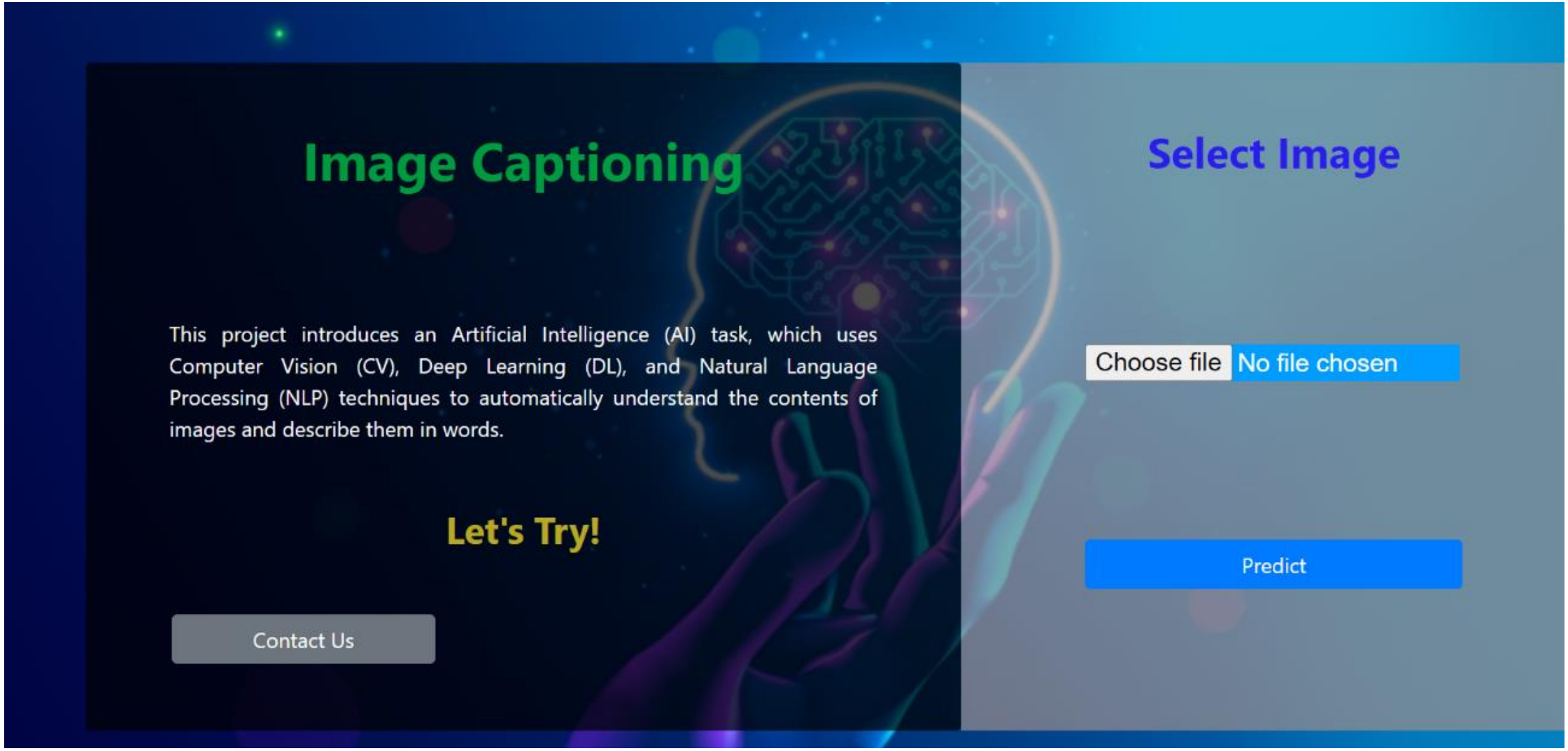
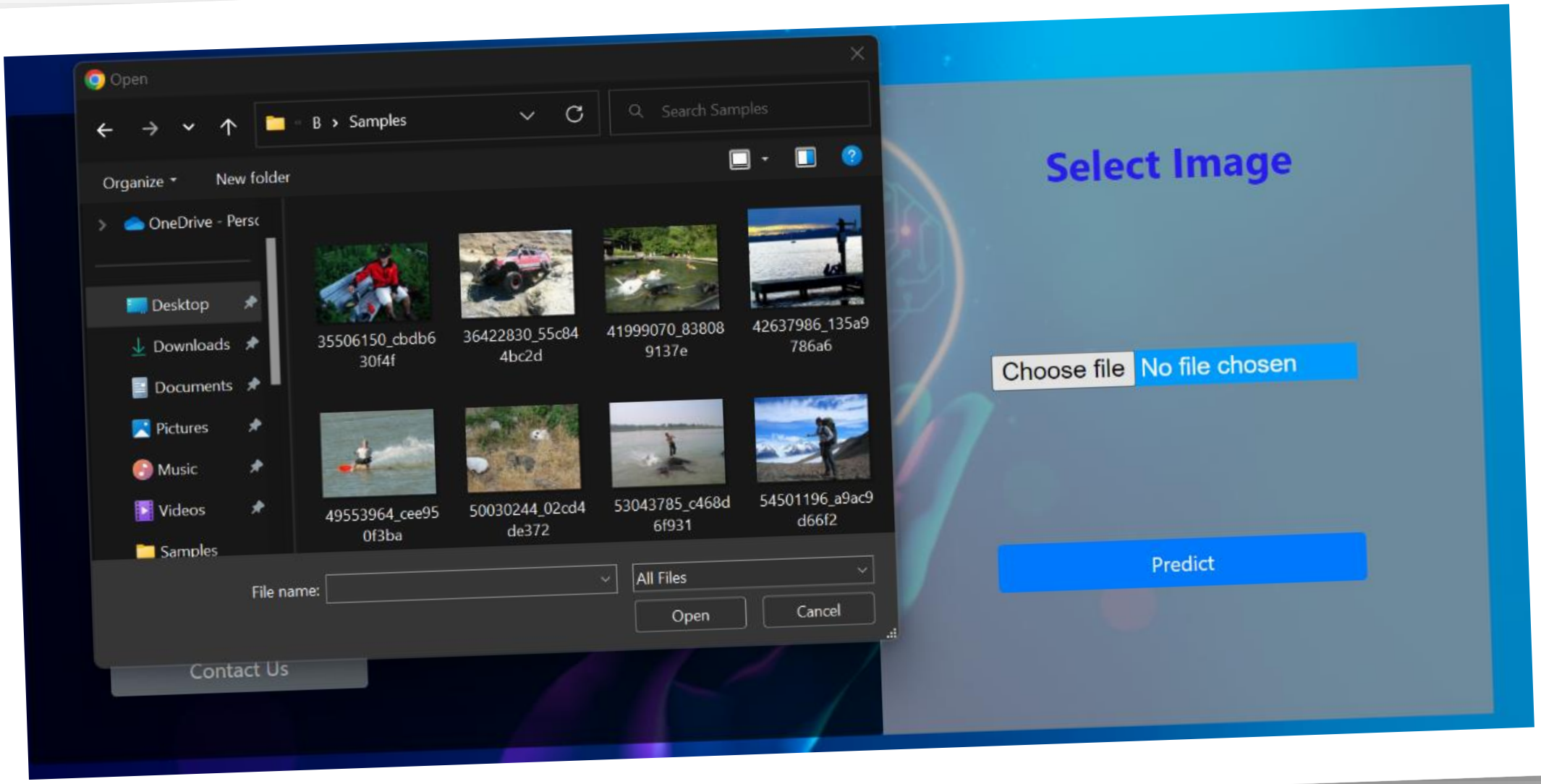


Image Captioning
This project introduces an Artificial Intelligence (AI) task, which uses Computer Vision (CV), Deep Learning (DL), and Natural Language Processing (NLP) techniques to automatically understand the contents of images and describe them in words.
Let's Try!
Contact Us
Select Image
Choose file
No file chosen
Predict


Open
B › Samples
Search Samples
Organize
New folder
OneDrive - Perso
Desktop
Downloads
Documents
Pictures
Music
Videos
Samples
35506150_cbdb630f4f
36422830_55c844bc2d
41999070_838089137e
42637986_135a9786a6
49553964_cee950f3ba
50030244_02cd4de372
53043785_c468d6f931
54501196_a9ac9d66f2
File name:
All Files
Open
Cancel
Contact Us
Select Image
Choose file
No file chosen
Predict

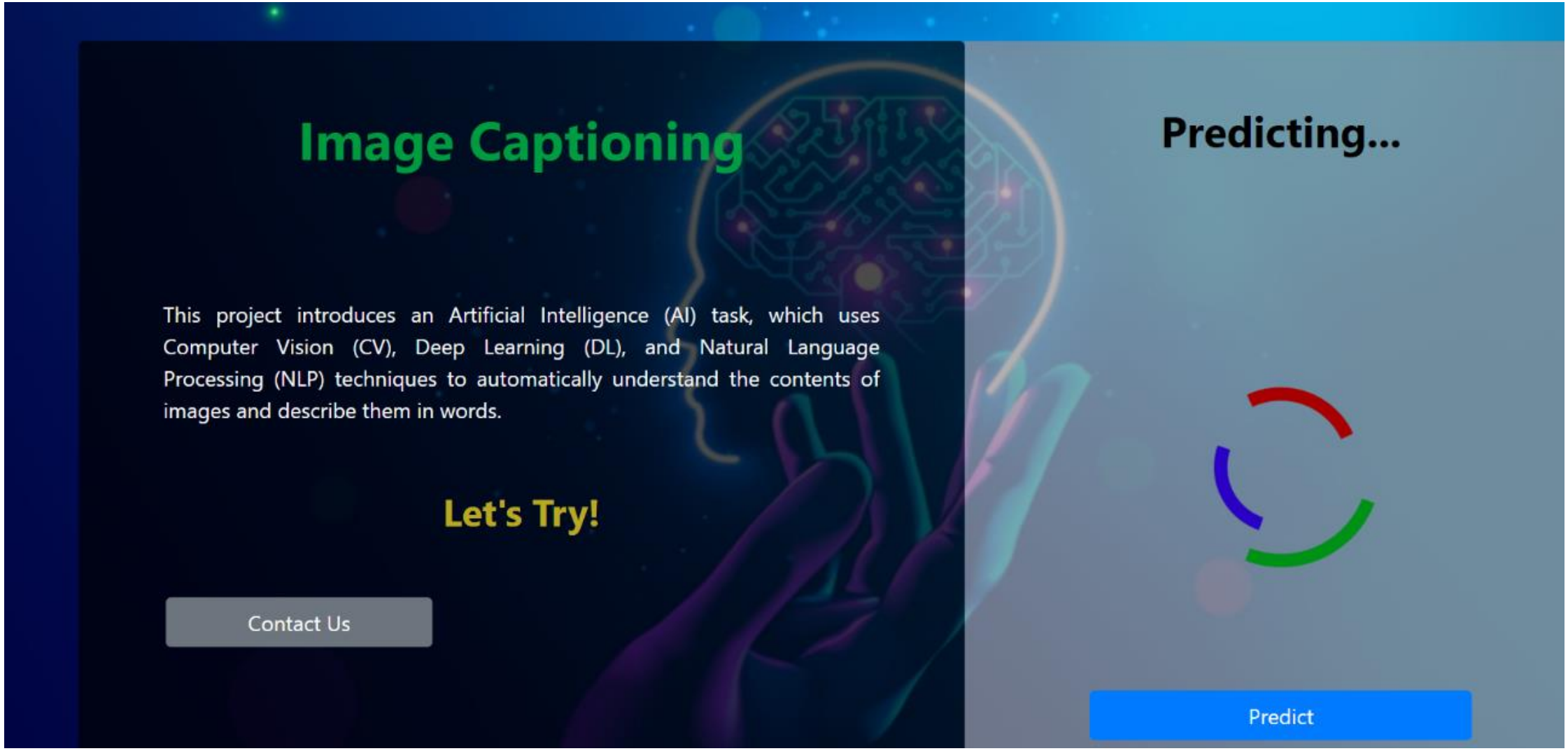
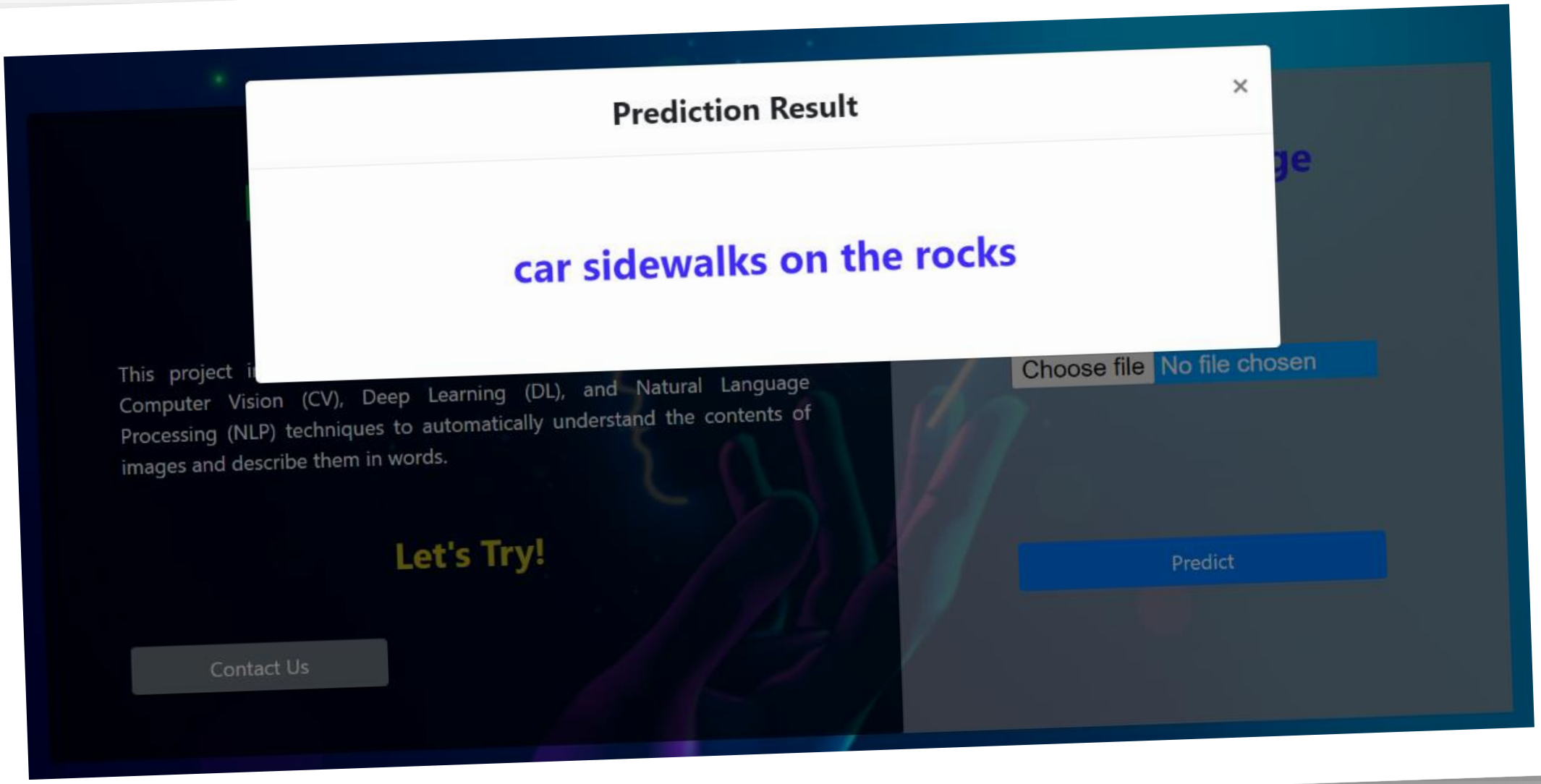


Image Captioning
This project introduces an Artificial Intelligence (AI) task, which uses Computer Vision (CV), Deep Learning (DL), and Natural Language Processing (NLP) techniques to automatically understand the contents of images and describe them in words.
Let's Try!
Contact Us
Predicting...
Predict


Prediction Result
car sidewalks on the rocks
This project i
Computer Vision (CV), Deep Learning (DL), and Natural Language Processing (NLP) techniques to automatically understand the contents of images and describe them in words.
Let's Try!
Contact Us
Choose file
No file chosen
Predict

Thank you so much!