



IMAGE RECOGNITION WITH IBM CLOUD VISUAL RECOGNITION

AGENDA

- ☐ INTRODUCTION
- ☐ IBM CLOUD
- ☐ DEVOLPMENT PART
- ✓ DESIGN THE WEB INTERFACE
- ✓ DEPLOY YOU APPLICATION
- ✓ OUTPUT
- ☐ AI-GENERATED CAPTIONS
- ☐ CONCLUSION

INTRODUCTION

Image recognition, also known as image detection or computer vision, is a technology that enables machines to interpret and understand visual information from images or videos. It involves the use of algorithms and deep learning models to identify objects, patterns, text, or any other relevant information within an image or video frame. Image recognition systems analyze and process visual data, making it possible for computers to "see" and comprehend the content of images similar to the way humans do.

IBM CLOUD

☐ **Create an IBM Cloud Account:**

Go to the IBM Cloud website (<https://cloud.ibm.com/>).

Sign up for a new account or log in if you already have one.

☐ **Create a Visual Recognition Service:**

After logging in, navigate to the IBM Cloud Catalog.

Search for "Visual Recognition" and select the service.

Choose a Lite plan and create the service instance.

Once the service is created, note down the API key and endpoint URL. You'll need these for authentication.

DEVELOPMENT PART

Continue building the image recognition system by integrating IBM cloud visual recognition and AI-generated captions

Step 1: Set up IBM Cloud Visual Recognition Service

- Sign in to your IBM Cloud account or create one if you haven't already.
- Go to the IBM Cloud Catalog and select the Visual Recognition service.
- Create an instance of the service and get the API key and endpoint URL.

Step 2: Integrate IBM Cloud Visual Recognition into your application

- Use the provided API key and endpoint URL to authenticate and access the Visual Recognition service.
- Utilize the provided SDKs or REST APIs to send images to the service for classification and recognition.
- Receive the results, which could include labels, classes, or other relevant information about the contents of the image.

Step 3: Integrate AI-generated captions

- Use the output from the IBM Cloud Visual Recognition service as input to your AI-generated captioning system.
- You can preprocess the output by selecting the most relevant labels or classes from the Visual Recognition results.
- Pass this refined data to your AI-generated captioning system for generating relevant captions for the images

Step 4: Display the results

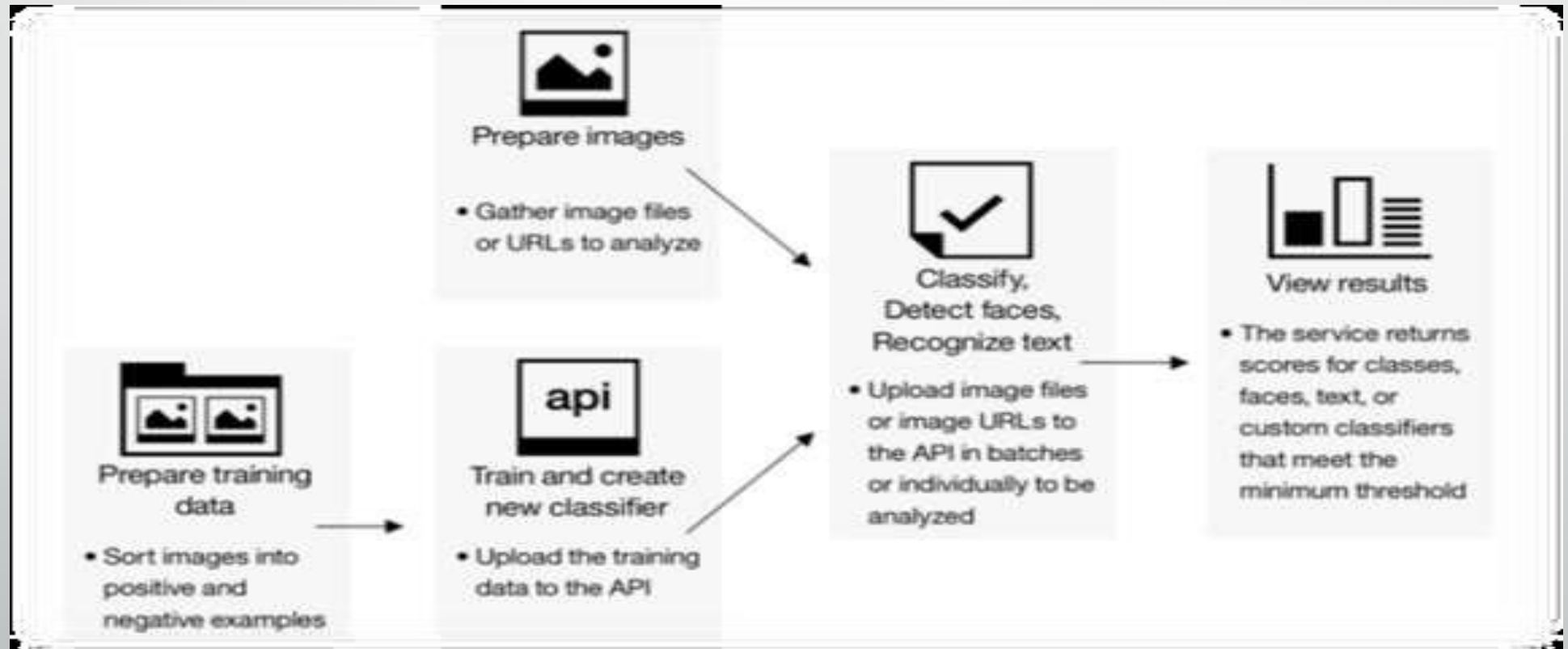
- Display the image along with the generated captions to provide meaningful context to the users.
- Ensure that the UI is intuitive and user-friendly, providing a seamless experience for the users to interact with the system.

Step 5: Testing and Iteration

- Test the integrated system thoroughly to ensure that it is functioning as expected.
- Gather feedback from users and incorporate any necessary changes or improvements to enhance the accuracy and
- relevance of the generated captions.

Step 6: Deployment and Maintenance

- Deploy the integrated system to your desired platform or application.
- Regularly monitor the system's performance and make necessary updates to keep the integration up-to-date with any changes in the IBM Cloud Visual Recognition service or your AI-generated captioning system.



DESIGN THE WEB INTERFACE

1.HTML and CSS:

Create an HTML file for the user interface.

Design a simple form allowing users to upload images.

Source file(index.html)

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Image Recognition</title>
<style>
/* Add your CSS styles here */
</style>
</head>
<body>
```



```
<h1>Image Recognition</h1>
<form id="upload-form">
  <input type="file" id="image-input" accept="image/*" required>
  <button type="submit">Upload</button>
</form>
<div id="result"></div>
<script src="script.js"></script>
</body>
</html>
```

2.JavaScript

Create a javascript file to handle form submission and API calls.

Source file(script.js)

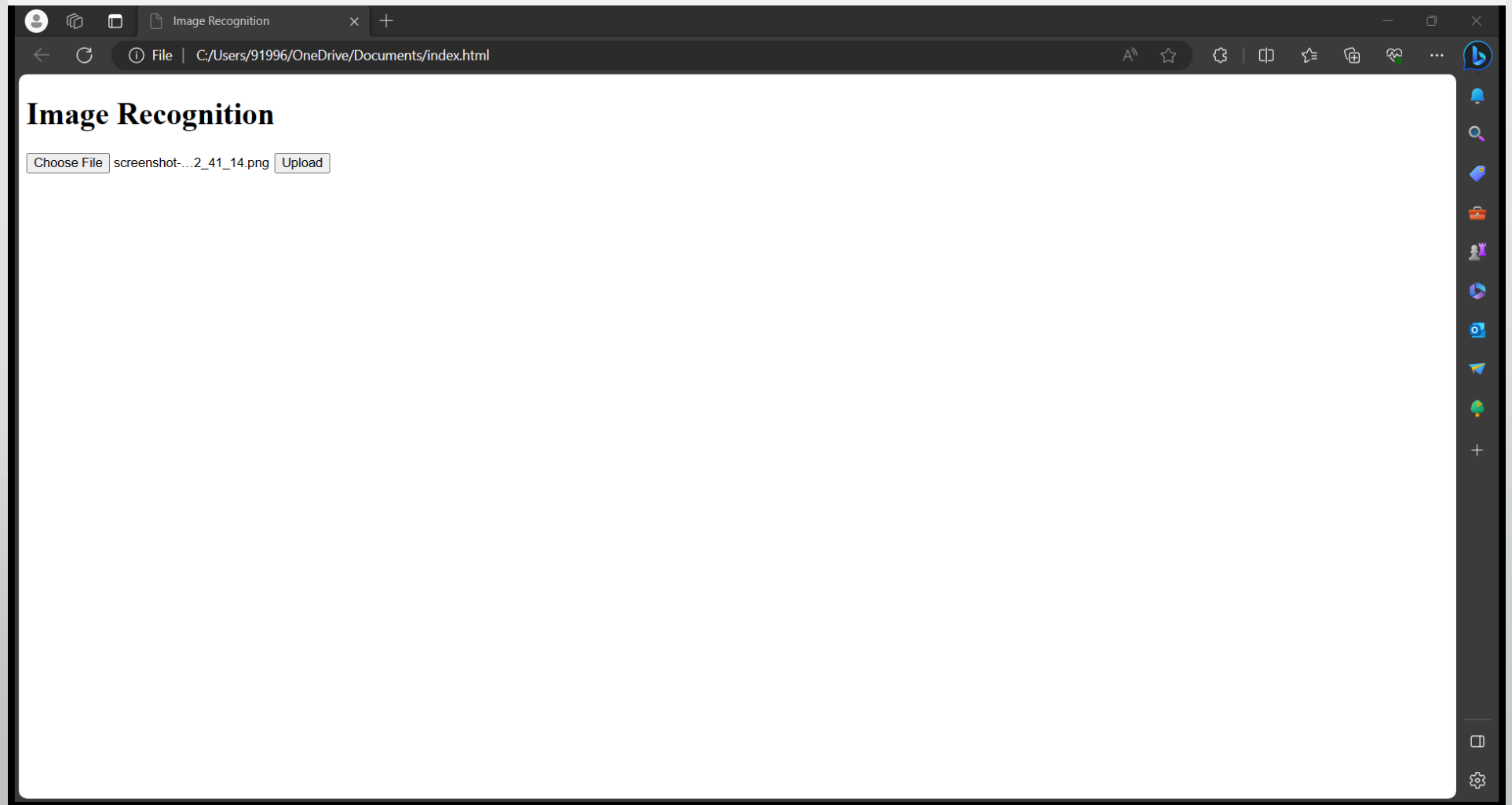
```
document.getElementById('uploadform').addEventListener('submit', function (event) {
  event.preventDefault();
  const formData = new FormData();
  const fileInput = document.getElementById('image-input');
```

```
formData.append('images_file', fileInput.files[0]);  
console.log('Before fetch request');  
fetch('/api/upload', {  
  method: 'POST',  
  body: formData  
})  
.then(response => response.json())  
.then(data => {  
  console.log('Fetch successful:', data);  
  document.getElementById('result').innerText = `Caption: ${data.caption}`;  
})  
.catch(error => console.error('Error:', error));  
});  
console.log('After fetch request');
```

DEPLOY YOUR APPLICATION

- Deploy your backend (Node.js) application to a hosting service like IBM Cloud, or any other platform of your choice.
- Upload your frontend (HTML, CSS, and JavaScript) files to the hosting service or use a static file hosting service.
- Now, users can access your web interface, upload images, and view the AI-generated captions using the IBM Cloud Visual Recognition service.

OUTPUT



AI-GENERATED CAPTIONS

Use natural language generation to create captions for the recognized images

- 1. Image Recognition: Use a pre-trained deep learning model for image recognition to identify objects, scenes, and context within an image. Popular frameworks like TensorFlow, PyTorch, and Keras offer pre-trained models that can be used for this purpose. Common models include VGG, ResNet, Inception, and Mobile Net.
- 2. Data Preprocessing: Once you have the image features extracted, preprocess them to make them suitable as input to the NLG model. This might involve reshaping, scaling, or other necessary transformations.

- Natural Language Generation: Utilize a suitable NLG model, such as GPT-3 or other language models, to generate captions or descriptions based on the recognized features of the image. Fine-tune the NLG model using a dataset of image-caption pairs to ensure it learns how to generate appropriate and relevant descriptions.
- 4. Post-Processing: After generating the captions, perform any necessary post-processing to refine the text, improve coherence, and ensure grammatical correctness. This might involve checking for redundancies, correcting grammar, or adjusting the tone to match the context.
- 5. Integration: Finally, integrate the image recognition and NLG components into a unified system. image recognition output serves as input to the NLG model, and the NLG output is correctly associated with the corresponding image.

CONCLUSION

Integrating IBM Cloud Visual Recognition with Ai-generated captions can lead to a powerful image recognition system that offers a comprehensive understanding of the visual content. By combining the robust image recognition capabilities of IBM Cloud Visual Recognition with the context-enhancing capabilities of AI-generated captions, the system can provide users with a more holistic and informative experience.