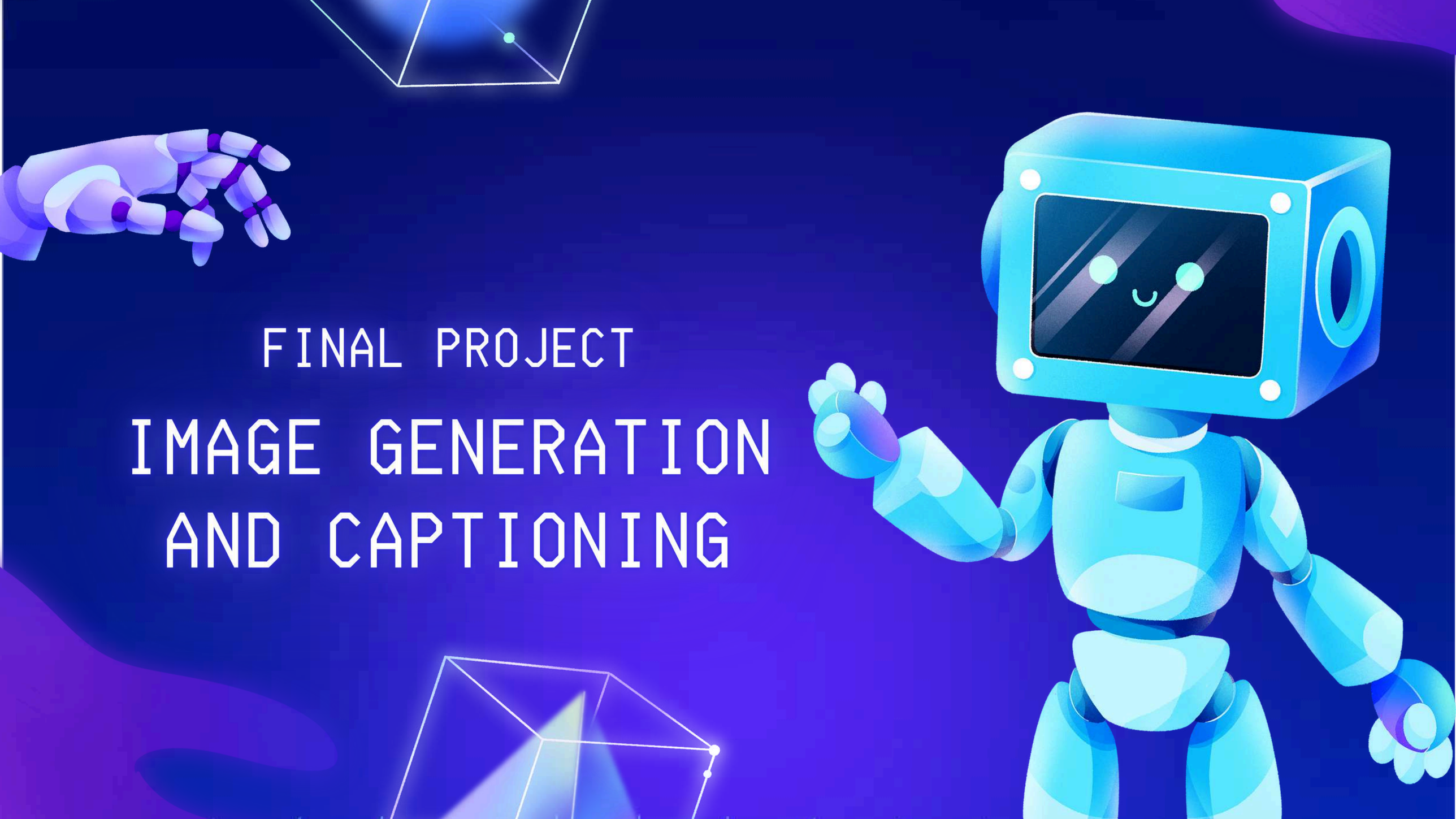
FINAL PROJECT

IMAGE GENERATION AND CAPTIONING



OUR TEAM

Shahad Alhassani

Data Analyst

Shahadbalhassani@gmail.com

@Shahad Alhassani

Noura Abdullah

Data Analyst

nourhamushi@gmail.com

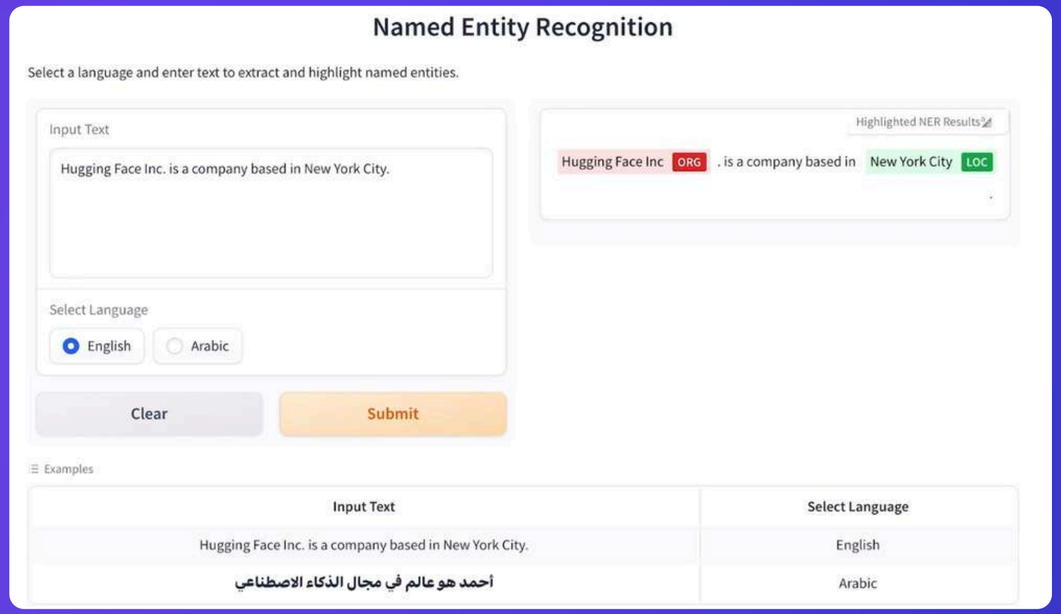
@noura abdullah

PREVIOUS WORK

This project aims to demonstrate named entity recognition (NER) for both English and Arabic languages. It utilizes Hugging Face Transformers and Gradio to extract and highlight named entities from the input text.

PROJECT LINK

PREVIOUS WORK



PROJECT CONCEPTS

To generate images based on captions from uploaded images and provide translations of those captions from English to Arabic.

IMAGE CAPTIONING USING BLIP

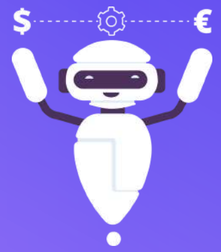


IMAGE GENERATION USING STABLE DIFFUSION

TRANSLATION USING NLLB

MODEL JUSTIFICATION

01

- BLIP (Image Captioning):
  1. Chosen for its state-of-the-art performance in generating descriptive captions based on images.

02

- Stable Diffusion (Image Generation):
  1. A widely used and powerful diffusion model for generating high-quality images based on textual prompts.

03

- NLLB-200 (Translation):
  1. A multilingual translation model, perfect for translating between English and Arabic, with efficient processing and high accuracy

PIPELINE WORKFLOW

01

Upload Image

The user uploads an image.

02

Caption Generation

Using BLIP, a caption in English is generated based on the image content.

03

Translation

The caption is translated from English to Arabic using the NLLB model.

04

Image Generation

Stable Diffusion generates new images based on the English caption.

05

Display

The original and translated captions are displayed alongside the generated images.

01

PIPELINE IMPLEMENTATION

Import for libraries

Gradio	For building the user interface.
Transformers	To load and apply models (BLIP and NLLB).
Diffusers	For Stable Diffusion pipeline.
Torch	For GPU acceleration.
Wget	For downloading the images.

```
!pip install gradio
!pip install transformers
!pip install diffusers
!pip install torch
```

PIPELINE IMPLEMENTATION

02

Define the device to use

```
# Define the device to use (either "cuda" for GPU or "cpu" for CPU)
device = "cuda" if torch.cuda.is_available() else "cpu"
```

03

Loading Models

```
# Load the models
caption_image = pipeline("image-to-text", model="Salesforce/blip-image-captioning-large", device=device)
sd_pipeline = StableDiffusionPipeline.from_pretrained("runwayml/stable-diffusion-v1-5").to(device)

# Load the translation model (English to Arabic)
translator = pipeline(
    task="translation",
    model="facebook/nllb-200-distilled-600M",
    torch_dtype=torch.bfloat16,
    device=device
)
```

04 PIPELINE IMPLEMENTATION

Download image

```
# Download the image
url1 = "https://github.com/Shahad-b/Image-database/blob/main/sea.jpg?raw=true"
sea = wget.download(url1)

url2 = "https://github.com/Shahad-b/Image-database/blob/main/Cat.jpeg?raw=true"
Cat = wget.download(url2)

url3 = "https://github.com/Shahad-b/Image-database/blob/main/Car.jpeg?raw=true"
Car = wget.download(url3)
```

05 PIPELINE IMPLEMENTATION

Main Function

```
# Function to generate images based on the image's caption
def generate_image_and_translate(image, num_images=1):
    # Generate caption in English from the uploaded image
    caption_en = caption_image(image)[0]['generated_text']

    # Translate the English caption to Arabic
    caption_ar = translator(caption_en, src_lang="eng_Latn", tgt_lang="arb_Arab")[0]['translation_text']

    generated_images = []

    # Generate the specified number of images based on the English caption
    for _ in range(num_images):
        generated_image = sd_pipeline(prompt=caption_en).images[0]
        generated_images.append(generated_image)

    # Return the generated images along with both captions
    return generated_images, caption_en, caption_ar
```

06

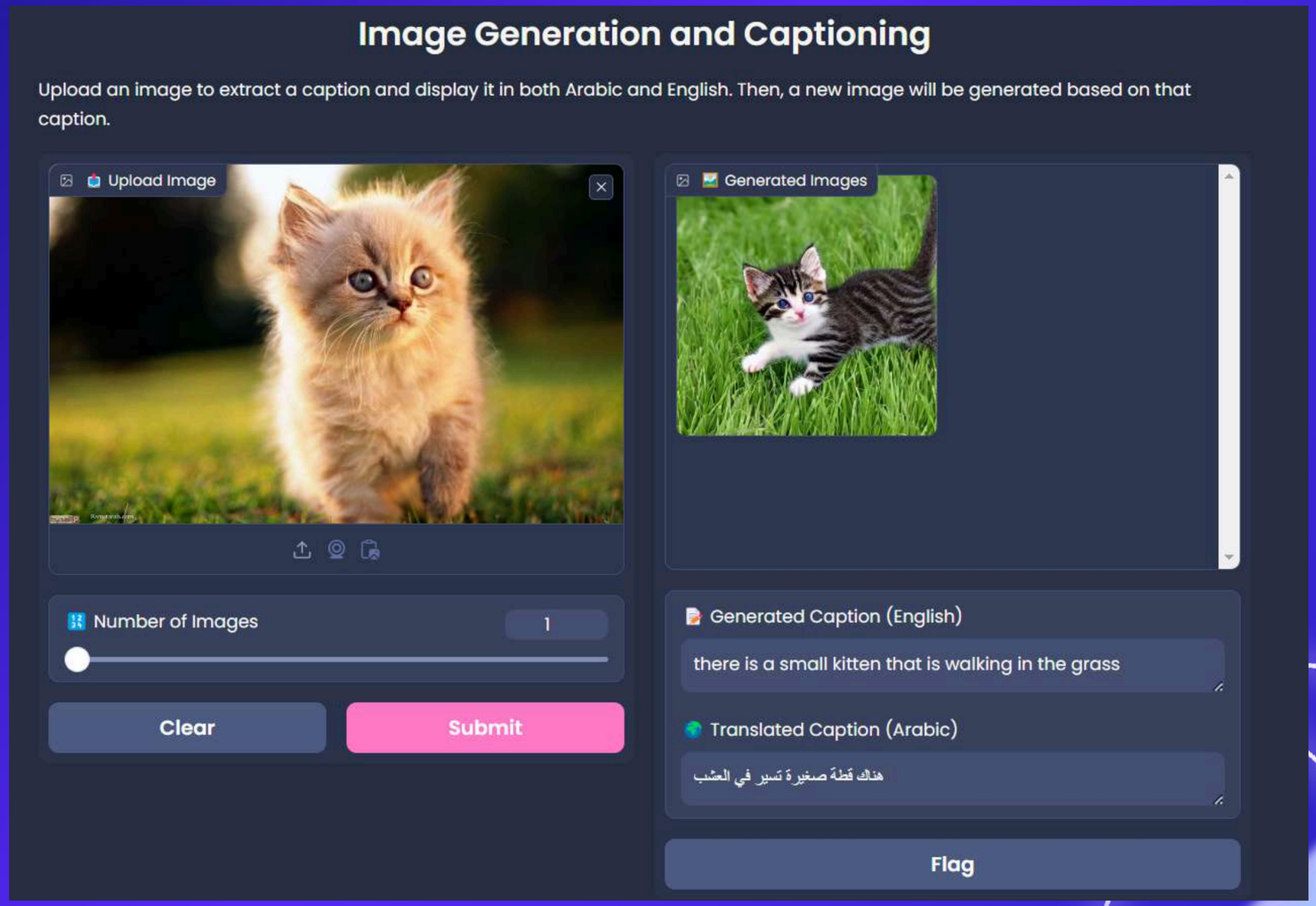
PIPELINE IMPLEMENTATION

Interface Setup

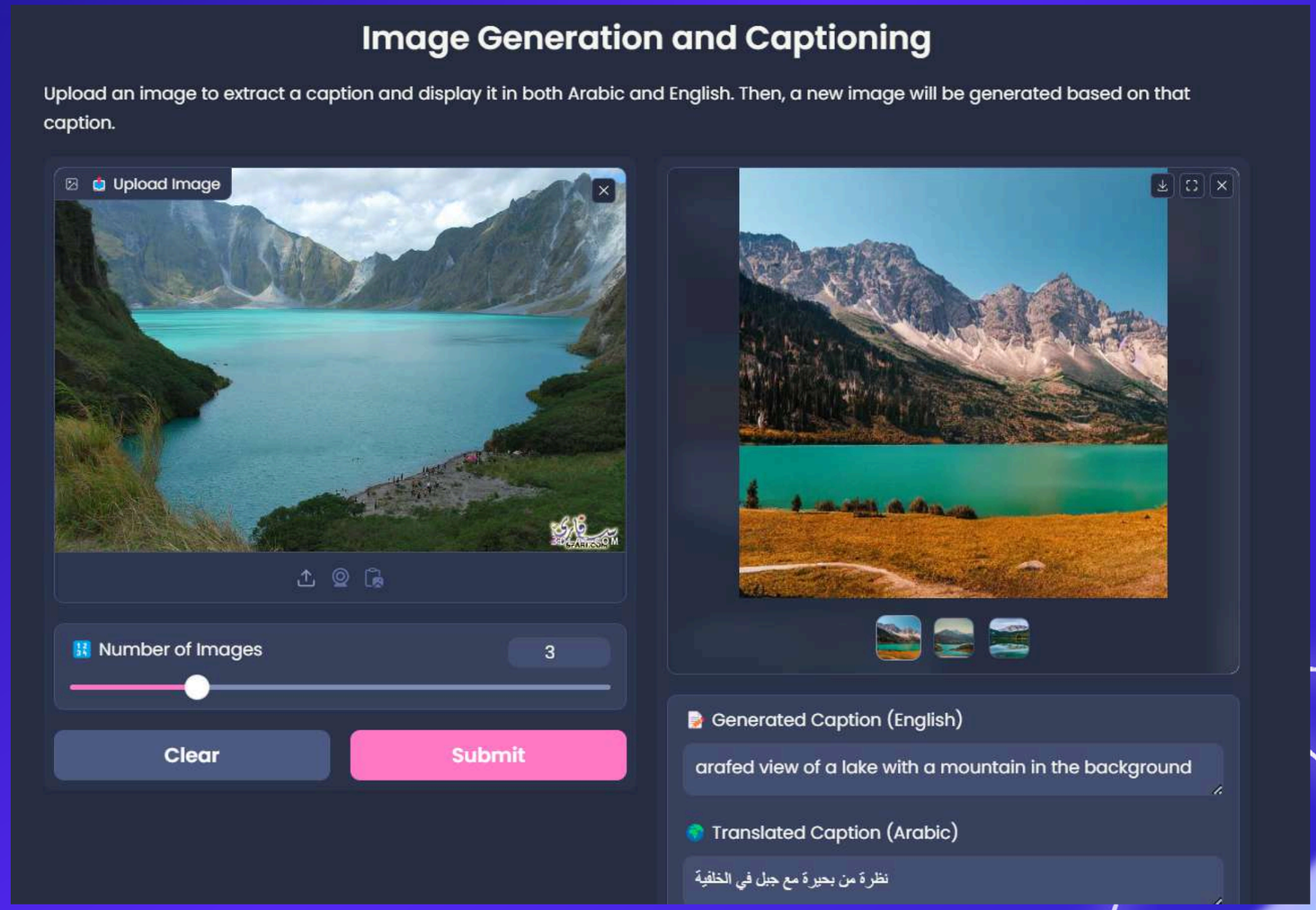
```
# Set up the Gradio interface
interface = gr.Interface(
    fn=generate_image_and_translate,   # Function to call when processing input
    inputs=[
        gr.Image(type="pil", label="📤 Upload Image"), # Input for image upload
        gr.Slider(minimum=1, maximum=10, label="🔢 Number of Images", value=1, step=1) # Slider to select number of images
    ],
    outputs=[
        gr.Gallery(label="🖼️ Generated Images"),
        gr.Textbox(label="📝 Generated Caption (English)", interactive=False),
        gr.Textbox(label="🌍 Translated Caption (Arabic)", interactive=False)
    ],
    title="Image Generation and Captioning", # Title of the interface
    description="Upload an image to extract a caption and display it in both Arabic and English. Then, a new image will be gene
    examples=[ # Example input
        ["sea.jpg", 3],
        ["Cat.jpeg", 4],
        ["Car.jpeg", 2]
    ],
    theme='freddyaboulton/dracula_revamped' # Determine theme
)

# Launch the Gradio application
interface.launch()
```

RESULTS & EXAMPLES



RESULTS & EXAMPLES



PROJECT ON COLAB

LINK

• LINKS

SHAHAD ALHASSANI

GITHUB 🔗 HUGGING FACE SPACE 🔗

NOURA ABDULLAH

GITHUB 🔗 HUGGING FACE SPACE 🔗

THANK YOU!