

Government College of Technology - Coimbatore

Transformative Image Captioning

PRESENTED BY : NALINI V

REGISTER NO : 71772117502

DEPARTMENT : COMPUTER SCIENCE AND ENGINEERING

PROJECT TITLE



Transformative Image Captioning: Harnessing
VisionEncoderDecoderModel and ViTImageProcessor



AGENDA

- ❖ Introduction to Image Captioning
- ❖ Problem Statement
- ❖ Project Overview
- ❖ End Users and Audience
- ❖ Solution and Value Proposition
- ❖ Key Features and Benefits
- ❖ Conclusion



PROBLEM STATEMENT

Challenge:

Inaccurate and Incomplete Image Descriptions

Goal:

Develop an AI-powered system for generating accurate and contextually relevant captions for images.



PROJECT OVERVIEW

- ❑ Introduction to VisionEncoderDecoderModel and ViTImageProcessor
- ❑ Key Components of the Project Pipeline:
- ❑ Image Loading and Preprocessing
- ❑ Feature Extraction
- ❑ Caption Generation
- ❑ Evaluation Metrics: BLEU Score, Accuracy, Relevance



WHO ARE THE END USERS?

- Individuals with Visual Impairments
- Content Creators and Publishers
- Image Search Engines and Platforms
- AI Researchers and Developers

YOUR SOLUTION AND ITS VALUE PROPOSITION



- ✓ Utilizing State-of-the-Art Transformer Models.
- ✓ Seamless Integration of Image Processing and Language Modeling.
- ✓ Accurate and Contextually Rich Image Captions.
- ✓ Enhanced Accessibility and User Experience.
- ✓ Potential for Improving Search Engine Optimization (SEO) for Visual Content.

THE WOW IN YOUR SOLUTION

AI-driven Accuracy: Generate Descriptive Captions with High Precision

Scalability: Process a Wide Range of Images with Efficiency

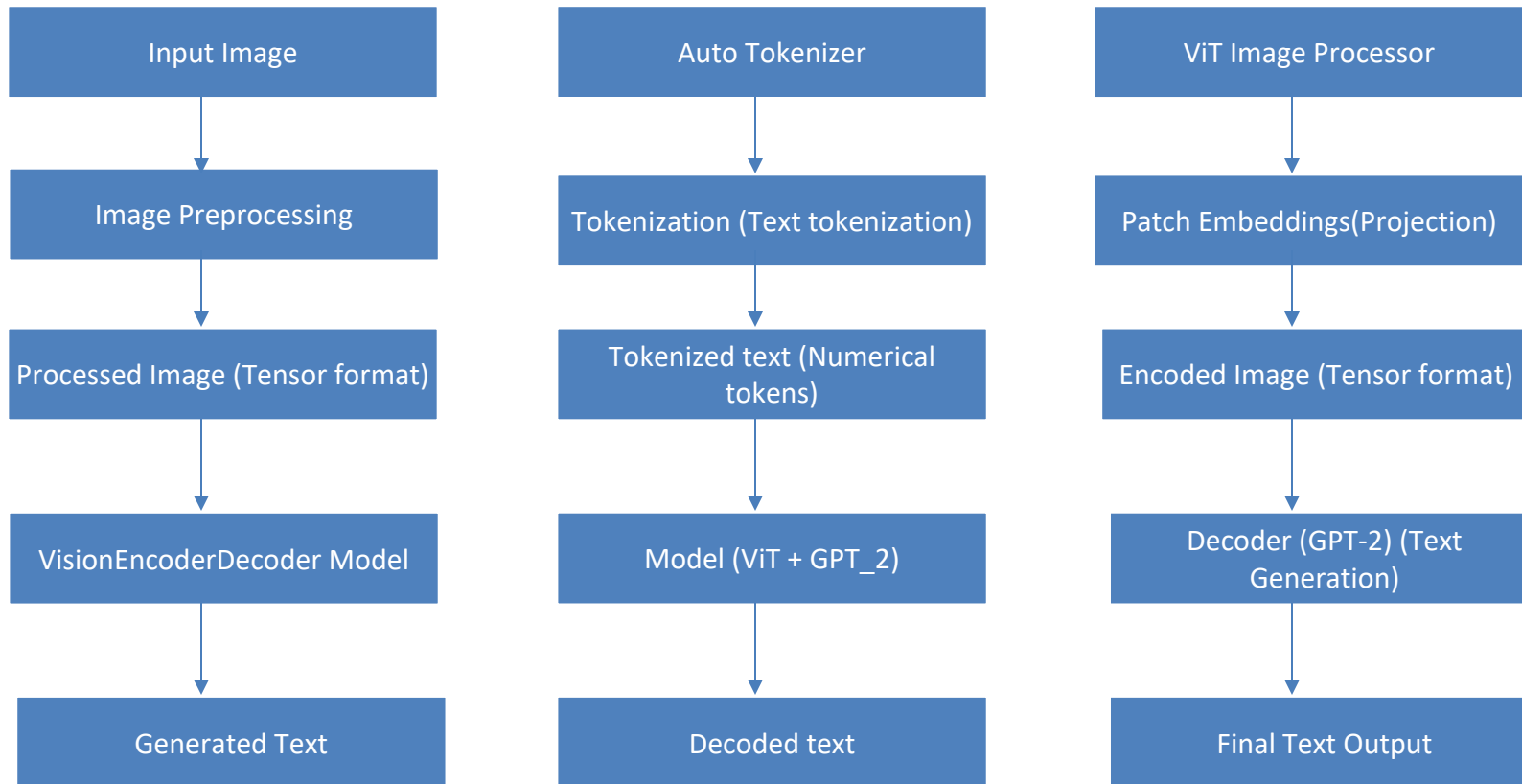
Customization: Fine-Tune the Model for Specific Domains or Use Cases

Real-World Impact: Improve Accessibility and User Interaction with Visual Content

Future Potential: Expand to Multimodal AI Applications for Comprehensive Content Understanding



MODELLING



RESULTS



```
def predict_step(image_paths):
    images = []
    for image_path in image_paths:
        i_image = Image.open(image_path)
        if i_image.mode != "RGB":
            i_image = i_image.convert(mode="RGB")
        images.append(i_image)

    pixel_values = feature_extractor(images=images, return_tensors="pt").pixel_values
    pixel_values = pixel_values.to(device)

    output_ids = model.generate(pixel_values, **gen_kwargs)

    preds = tokenizer.batch_decode(output_ids, skip_special_tokens=True)
    preds = [pred.strip() for pred in preds]
    return preds

predict_step(['/content/dogandcatimage.jpg'])
```

We strongly recommend passing in an 'attention_mask' since your input_ids may be padded. See <https://huggingface.co/docs/transformers/troubleshooting#incorrect-output>. You may ignore this warning if your 'pad_token_id' (50256) is identical to the 'bos_token_id' (50256), or the 'sep_token_id' (None), and your '['a brown and white dog and a brown and white cat']