



# Image To Text

---

Mateusz Jamróz  
Gerard Grzyb

# Image To Text

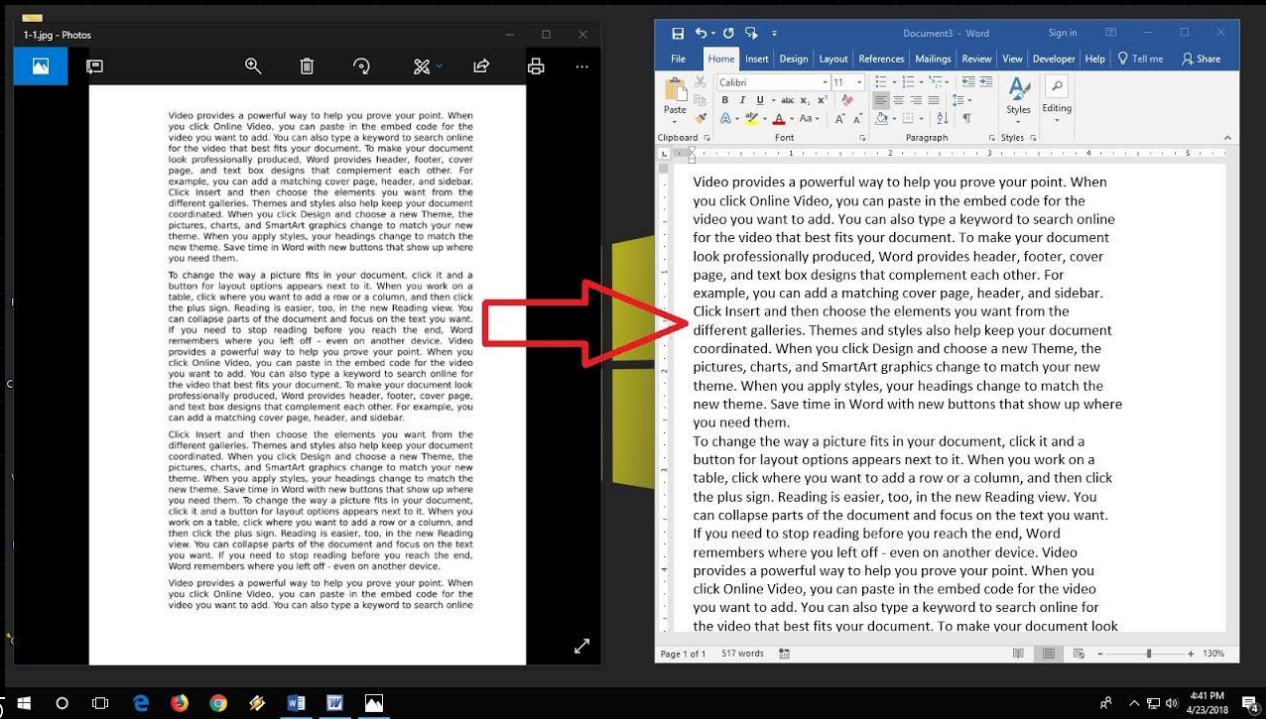
## 01

Przetwarzanie zdjęć na tekst

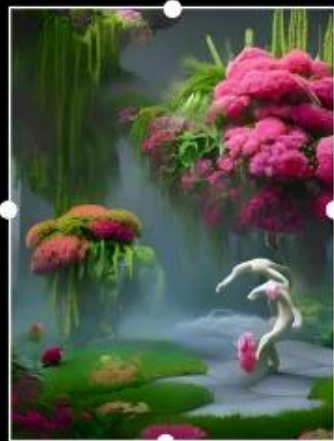
---



# Przetwarzanie zdjęć na tekst



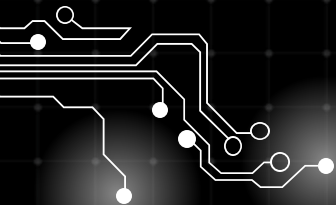
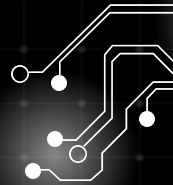
# Przetwarzanie zdjęć na tekst



ai

A surrealistic garden with extraordinary flowers

# Przetwarzanie zdjęć na tekst



# Przetwarzanie zdjęć na tekst

```
model1 = VisionEncoderDecoderModel.from_pretrained("nlpconnect/vit-gpt2-image-captioning")
feature_extractor1 = ViTImageProcessor.from_pretrained("nlpconnect/vit-gpt2-image-captioning")
tokenizer1 = AutoTokenizer.from_pretrained("nlpconnect/vit-gpt2-image-captioning")

device1 = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model1.to(device1)

max_length = 16
num_beams = 4
gen_kwargs = {"max_length": max_length, "num_beams": num_beams}

def image_to_text_model_1(image_url):
    raw_image = Image.open(requests.get(image_url, stream=True).raw).convert('RGB')

    pixel_values = feature_extractor1(images=[raw_image], return_tensors="pt").pixel_values
    pixel_values = pixel_values.to(device1)

    output_ids = model1.generate(pixel_values, **gen_kwargs)

    preds = tokenizer1.batch_decode(output_ids, skip_special_tokens=True)
    preds = [pred.strip() for pred in preds]
    return preds

def bytes_to_text_model_1(bts):
    pixel_values = feature_extractor1(images=[bts], return_tensors="pt").pixel_values
    pixel_values = pixel_values.to(device1)

    output_ids = model1.generate(pixel_values, **gen_kwargs)

    preds = tokenizer1.batch_decode(output_ids, skip_special_tokens=True)
    preds = [pred.strip() for pred in preds]
    print(preds[0])
```

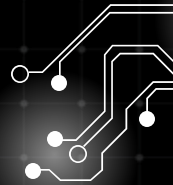
# Image To Text

01

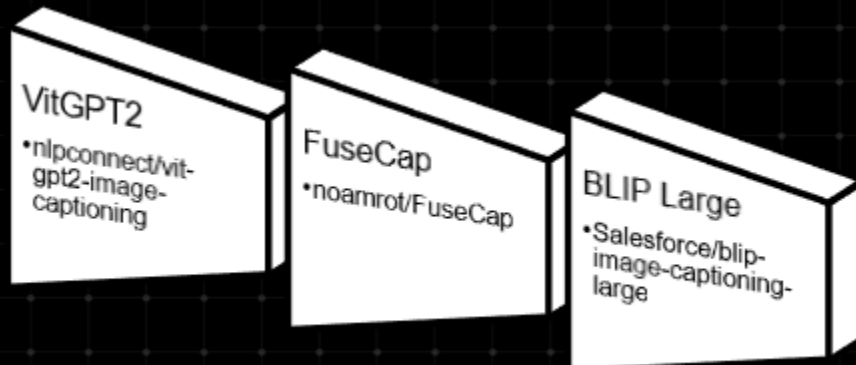
Przetwarzanie zdjęć na tekst

02

Wykorzystane modele



# Wykorzystane modele





# Image To Text

01

Przetwarzanie zdjęć na tekst

02

Wykorzystane modele

03

Zastosowanie w filmach



# Zastosowanie w filmach

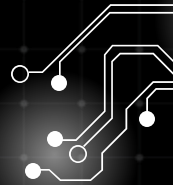
Podział filmu na  
klatki



Analiza każdej  
klatki jako zdjęcie



# Image To Text



01

Przetwarzanie zdjęć na tekst

02

Wykorzystane modele

03

Zastosowanie w filmach

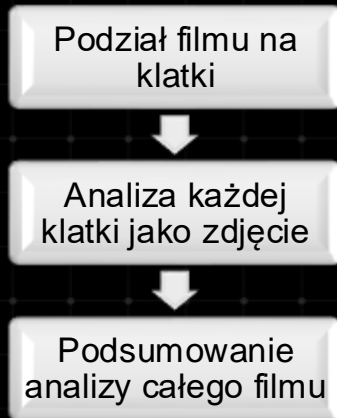
---

04

Sumaryzator



# Summaryzator



```
import cv2
from transformers import pipeline

summarizer = pipeline("summarization", model="facebook/bart-large-cnn")

> def FrameCapture(path): ...

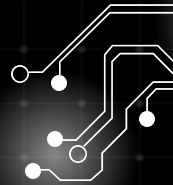
def summarize_articles(article1_content, article2_content, article3_content):
    article1_content = list(set([x for x in article1_content if x]))
    content1 = " ".join(article1_content)
    summary1 = summarizer(content1, max_length=130, min_length=30, do_sample=False)
    print("Summary for Article 1:")
    print(summary1)

    article2_content = list(set([x for x in article2_content if x]))
    print(article2_content)
    content2 = " ".join(article2_content)
    summary2 = summarizer(content2, max_length=130, min_length=30, do_sample=False)
    print("Summary for Article 2:")
    print(summary2)

    article3_content = list(set([x for x in article3_content if x]))
    content3 = " ".join(article3_content)
    summary3 = summarizer(content3, max_length=130, min_length=30, do_sample=False)
    print("Summary for Article 3:")
    print(summary3)
```



# Image To Text



01

Przetwarzanie zdjęć na tekst

02

Wykorzystane modele

03

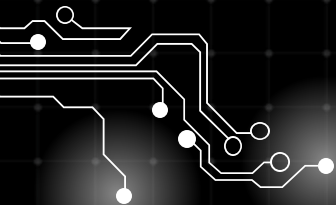
Zastosowanie w filmach

04

Sumaryzator

05

Porównanie wyników



# Porównanie wyników



# Porównanie wyników



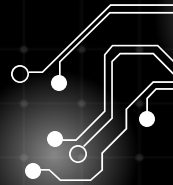


# Porównanie wyników





# Image To Text



01

Przetwarzanie zdjęć na tekst

02

Wykorzystane modele

03

Zastosowanie w filmach

04

Sumaryzator

05

Porównanie wyników

06

Wdrożenie na Spaces



# Wdrożenie na Spaces

## Video Captioning with Multiple Models 🤖

Upload a video to generate captions for its frames using three different models.

📁 Upload Video



Przeciągnij tutaj video

- lub -

Kliknij, aby przesłać



Generated Captions

Generate Captions

Use via API 🔥 · utworzone z gradio 🍌



# Live Demo



there is a small wooden door in the middle of a forest

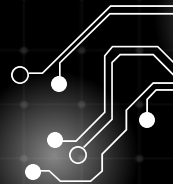


# Bibliografia

<https://opencv.org/>

[https://huggingface.co/models?pipeline\\_tag=text-to-speech](https://huggingface.co/models?pipeline_tag=text-to-speech)

<https://www.youtube.com/>



Dziękujemy za uwagę

