AI For Digital Collections

MetaMoscaic Data Flow

1) Data Source:

Start process where images are stored

Possible locations ↓
- Discovery Cluster
- S3 bucket

2) Image Processor:

Class to process images before they are fed to the model

class vars

filepath – where file is stored on S3 Bucket

discuss whether at this step you want to detect what file type it is, or assume it is a TIF file

Methods	Description
process_image →	takes TIF file converts it to a JPG stored at some local location where script.py file resides

↙ REDUCE SIZE

Return: jpg filepath

base_64_encode() → Encodes image into base_64 for Claude model

Return: base_64 encoding

gemini_upload() → genai_upload (new jpg file path)

3) Transcription Model - interface that Claude and Gemini scripts inherit

Class that specifically transcribes the back image

Methods

generate_transcription(img_back_filepath):
returns transcription returned from the model ← string

ask about more than 1 photographer ↘

extract_names()
returns photographername ← string

extract_dates()
returns list of dates ← list

extract_raw()
returns raw transcription

Constructor_init_: initializes prompt

get_token_size() ← depends on inherited classes not included in interface
returns num_tokens

4) Title Model - interface

Methods

generate_title(img_front_filepath)
returns generated title

Constructor_init_: initializes prompt

get_ tokens

returns tokens

5) Abstract Model - interface

Methods

generate_abstract(img_front_filepath)

returns generated abstract

Constructor_init_: initializes prompt

get_ tokens

returns tokens

6) CSV Writer

combines all metadata and writes it to a csv_file

Discuss where output should go

Class_vars

image_file_name
title
abstract
photographer_name
primary_date

methods:

write_to_csv(csv_file_path)

generate_json

↑ generates json version of

secondary-date

raw_transcription

total_tokens

metadata