LOGO-istics Final Presentation

Sam Parsons

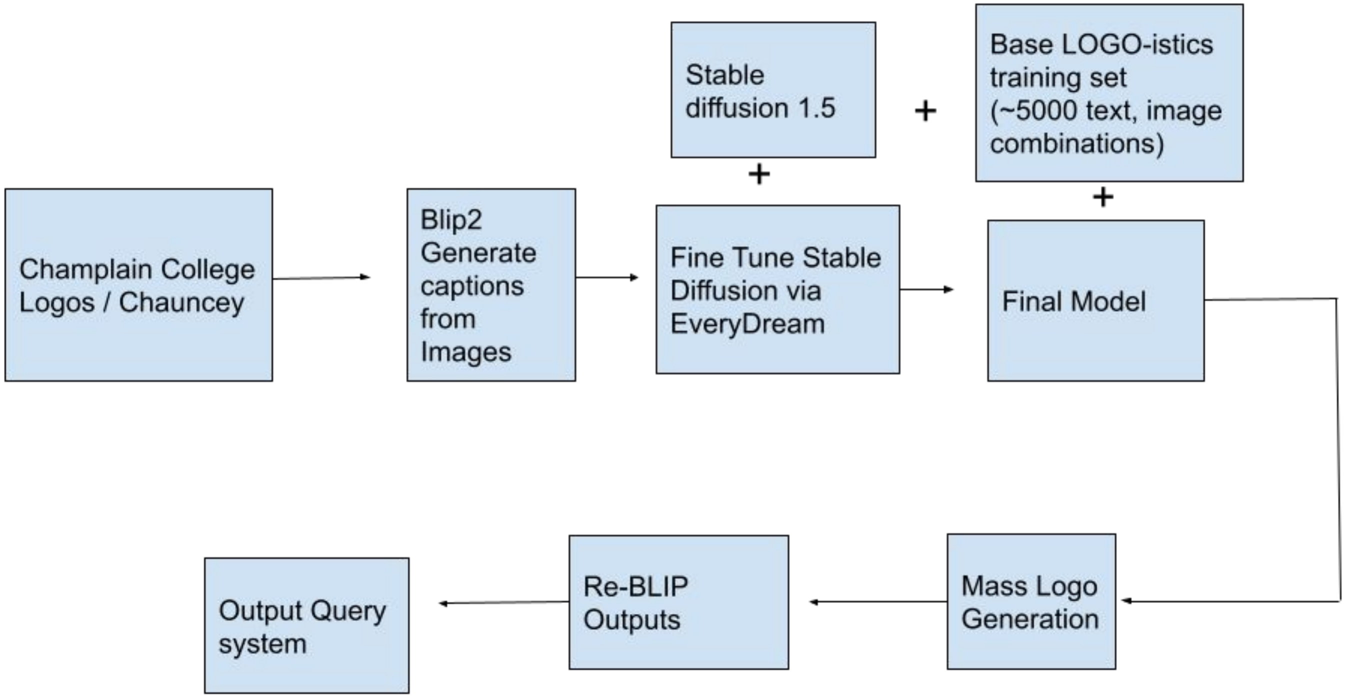
LOGO-istics: Ai Generated
Logos and Promotional Images

Abstract

“The logo is a visual representation of a company’s values and products. Through imagery/stylized text a company, like Nike, can make a logo that will invoke emotion in a consumer to become a lifelong customer. The design process of creating the perfect logo for a business is a long and arduous task. A strong well-designed logo can make all the difference in the world for a business and yet many do the bare minimum in designing their logo. This project aims to automate the process of designing a logo by using stable diffusion to create an AI-generated image.”

-Sam Parsons, LOGO-istics founder and CEO

LOGO-istics Flow Chart



Champlain College Logos / Chauncey

Blip2 Generate captions from Images

Stable diffusion 1.5

+

Fine Tune Stable Diffusion via EveryDream

+

Base LOGO-istics training set (~5000 text, image combinations)

+

Final Model

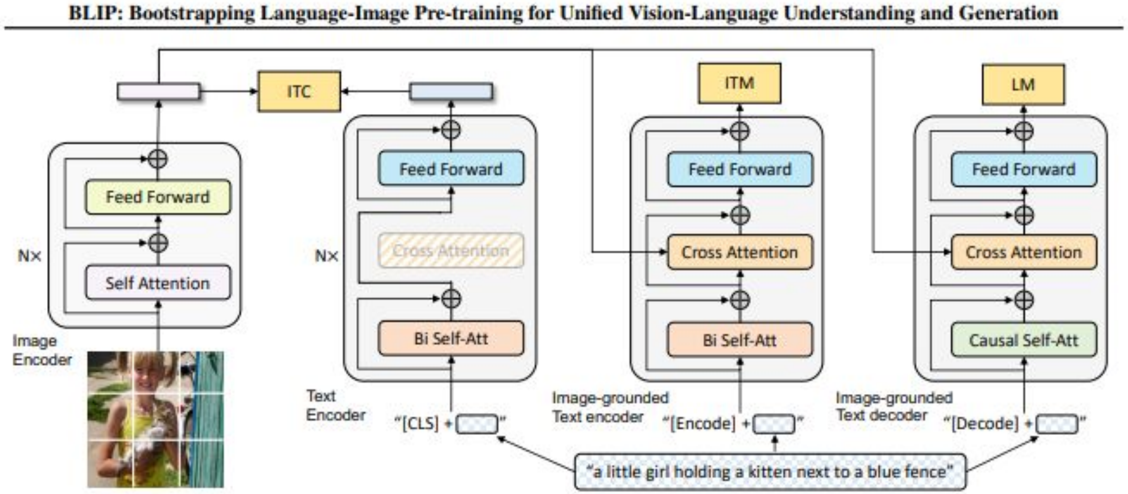
Mass Logo Generation

Re-BLIP Outputs

Output Query system

BLIP on a High Level

- Visual Transformer to encode image as embeddings
- Series of encoding and decoding
- ITC, ITM, LM
- CaptFit: Captioning and filtering

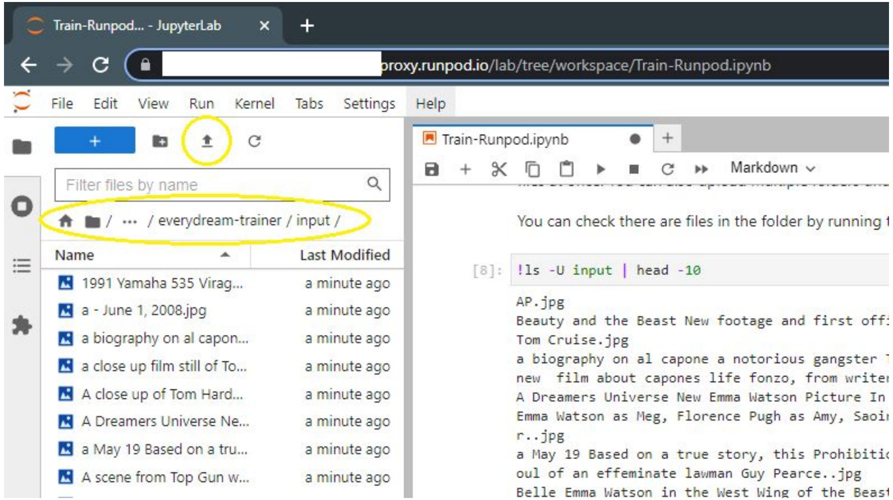


BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation

LOGO-istics: Ai Generated Logos and Promotional Images

Every Dream Trainer

- General purpose fine tuning software
- Meant for projects with their own curated data
- Introduces more training images by looping in LAION
- Checkpoints tested against prompts during training process
- Low cost easy setup



[redacted]proxy.runpod.io/lab/tree/workspace/Train-Runpod.ipynb

/ ... / everydream-trainer / input /

Name	Last Modified
1991 Yamaha 535 Virag...	a minute ago
a - June 1, 2008.jpg	a minute ago
a biography on al capon...	a minute ago
a close up film still of To...	a minute ago
A close up of Tom Hard...	a minute ago
A Dreamers Universe Ne...	a minute ago
a May 19 Based on a tru...	a minute ago
A scene from Top Gun w...	a minute ago

You can check there are files in the folder by running t

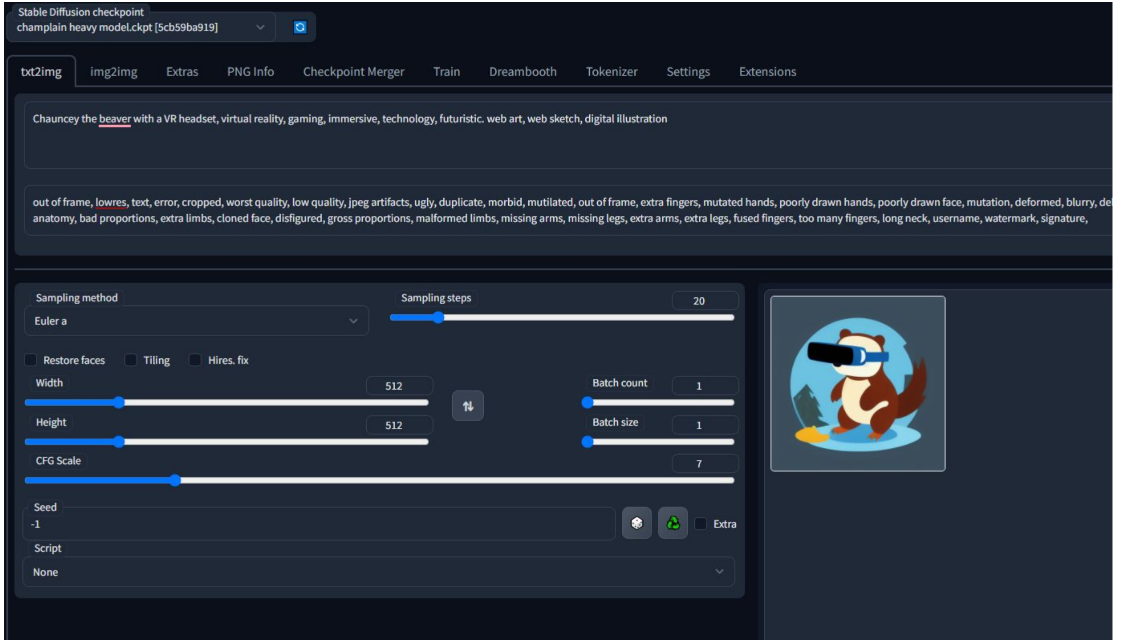
```
[8]: !ls -U input | head -10
```

```
AP.jpg
Beauty and the Beast New footage and first offi
Tom Cruise.jpg
a biography on al capone a notorious gangster T
new  film about capones life fonzo, from writer
A Dreamers Universe New Emma Watson Picture In
Emma Watson as Meg, Florence Pugh as Amy, Saoir
r..jpg
a May 19 Based on a true story, this Prohibitio
oul of an effeminate lawman Guy Pearce..jpg
Belle Emma Watson in the West Wing of the Beast
```

LOGO-istics: Ai Generated Logos and Promotional Images

Stable Diffusion

- Used Automatic1111 for stable diffusion
- Combining model checkpoints
- Pass in list of prompts to generate mass outputs
- Easily play around with configs



Stable Diffusion checkpoint: champlain heavy model.ckpt [5cb59ba919]

Chauncey the beaver with a VR headset, virtual reality, gaming, immersive, technology, futuristic. web art, web sketch, digital illustration

out of frame, lowres, text, error, cropped, worst quality, low quality, jpeg artifacts, ugly, duplicate, morbid, mutilated, out of frame, extra fingers, mutated hands, poorly drawn hands, poorly drawn face, mutation, deformed, blurry, de
anatomy, bad proportions, extra limbs, cloned face, disfigured, gross proportions, malformed limbs, missing arms, missing legs, extra arms, extra legs, fused fingers, too many fingers, long neck, username, watermark, signature,

Sampling method: Euler a

Sampling steps: 20

□ Restore faces □ Tiling □ Hires. fix

Width: 512

Height: 512

Batch count: 1

Batch size: 1

CFG Scale: 7

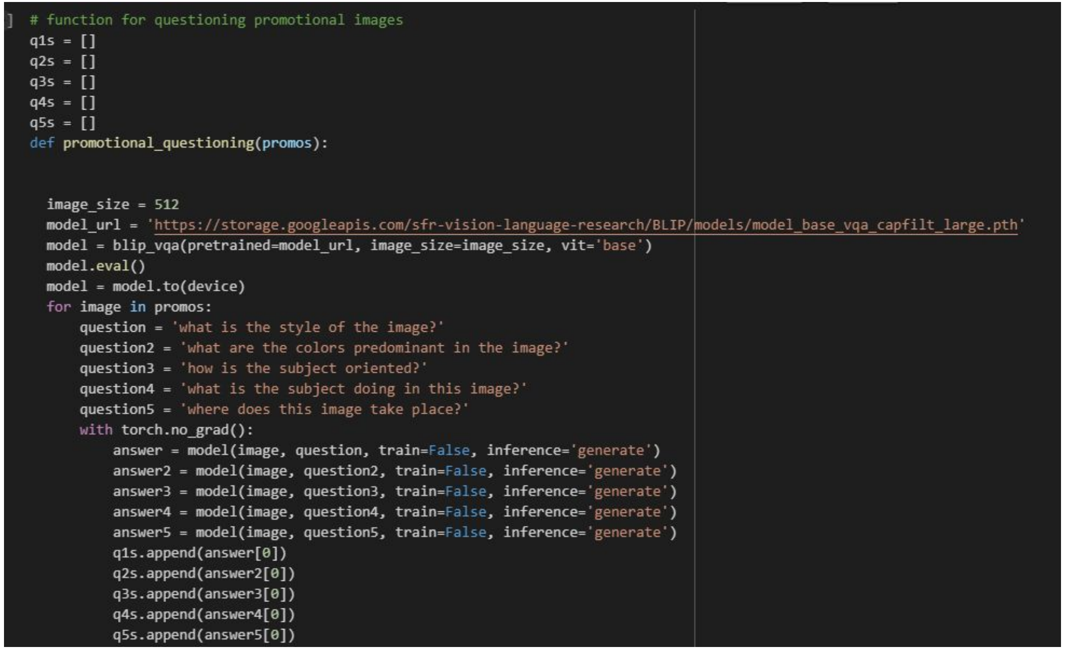
Seed: -1

□ Extra

Script: None

Re-BLIP Outputs

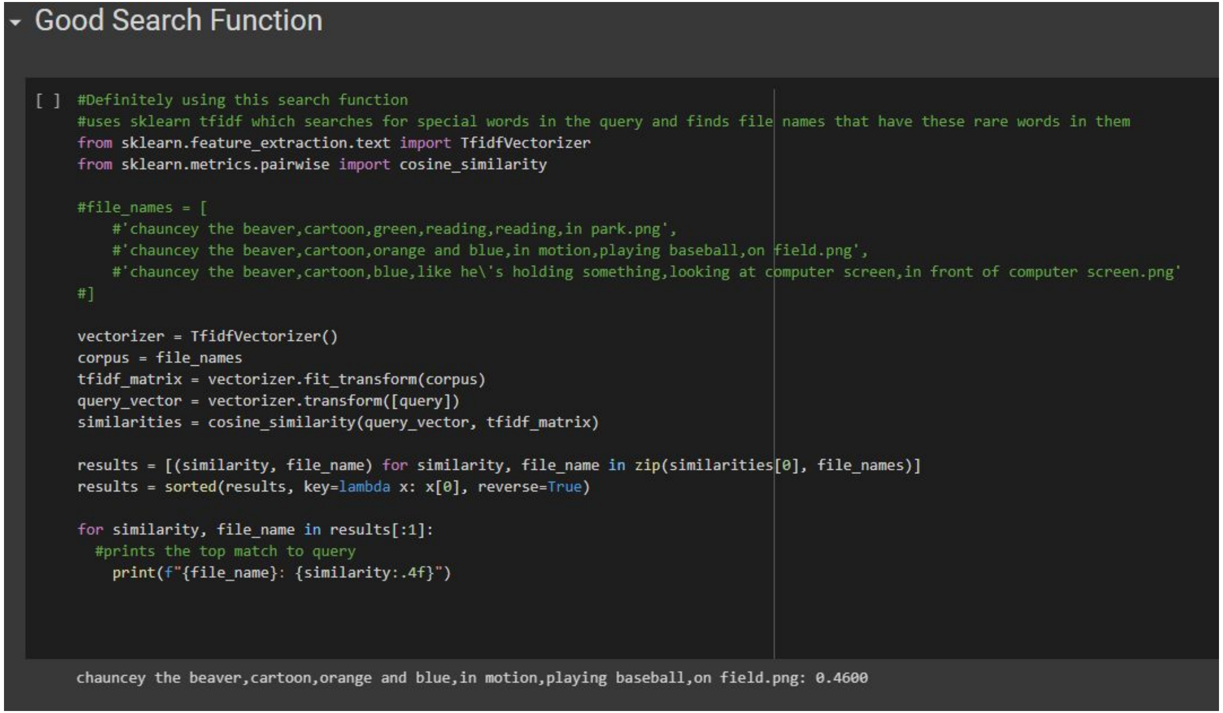
- Read in images and separate into promotional, logos, and misc
- Question the images
- Combine answers to make the text caption
- Rename file in google drive



```python
# function for questioning promotional images
q1s = []
q2s = []
q3s = []
q4s = []
q5s = []
def promotional_questioning(promos):


  image_size = 512
  model_url = 'https://storage.googleapis.com/sfr-vision-language-research/BLIP/models/model_base_vqa_capfilt_large.pth'
  model = blip_vqa(pretrained=model_url, image_size=image_size, vit='base')
  model.eval()
  model = model.to(device)
  for image in promos:
      question = 'what is the style of the image?'
      question2 = 'what are the colors predominant in the image?'
      question3 = 'how is the subject oriented?'
      question4 = 'what is the subject doing in this image?'
      question5 = 'where does this image take place?'
      with torch.no_grad():
          answer = model(image, question, train=False, inference='generate')
          answer2 = model(image, question2, train=False, inference='generate')
          answer3 = model(image, question3, train=False, inference='generate')
          answer4 = model(image, question4, train=False, inference='generate')
          answer5 = model(image, question5, train=False, inference='generate')
          q1s.append(answer[0])
          q2s.append(answer2[0])
          q3s.append(answer3[0])
          q4s.append(answer4[0])
          q5s.append(answer5[0])
```

Searching the Outputs

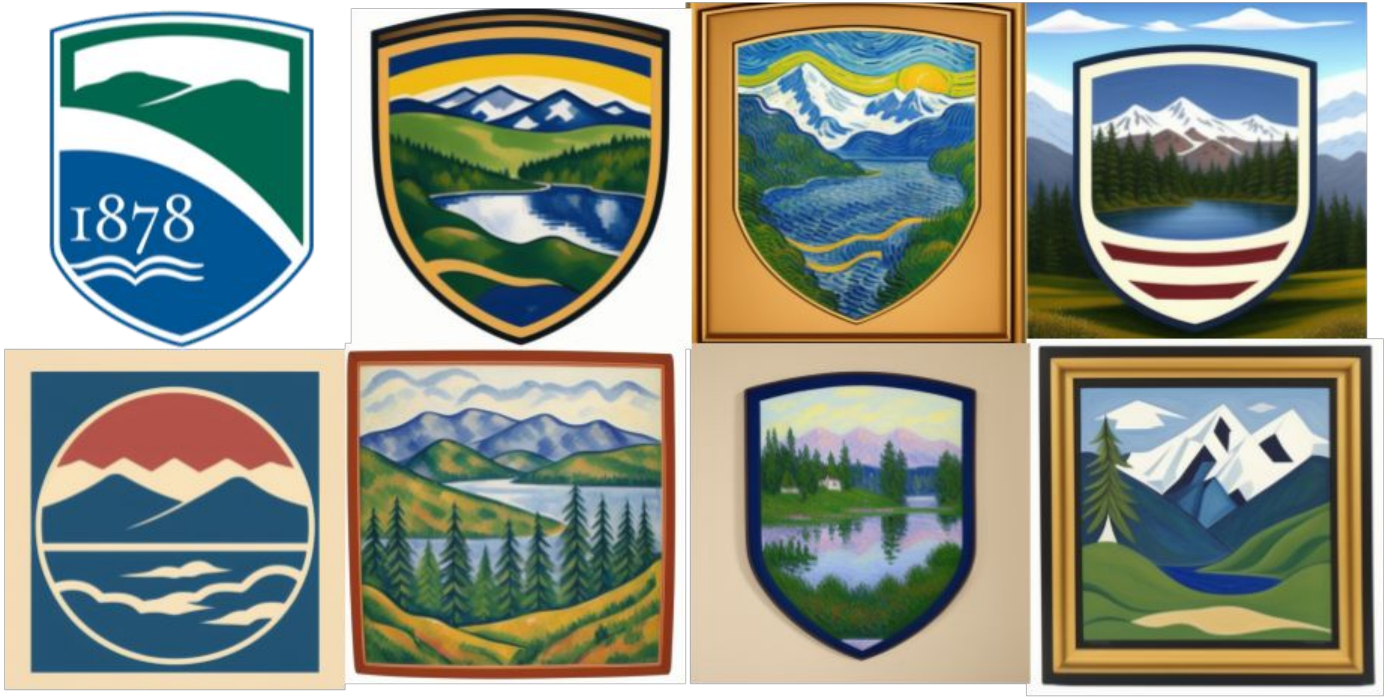
- Search problem with many similar names
- Used Sklearn TFIDF
- Rare words weighted heavier
- Less emphasis on common words



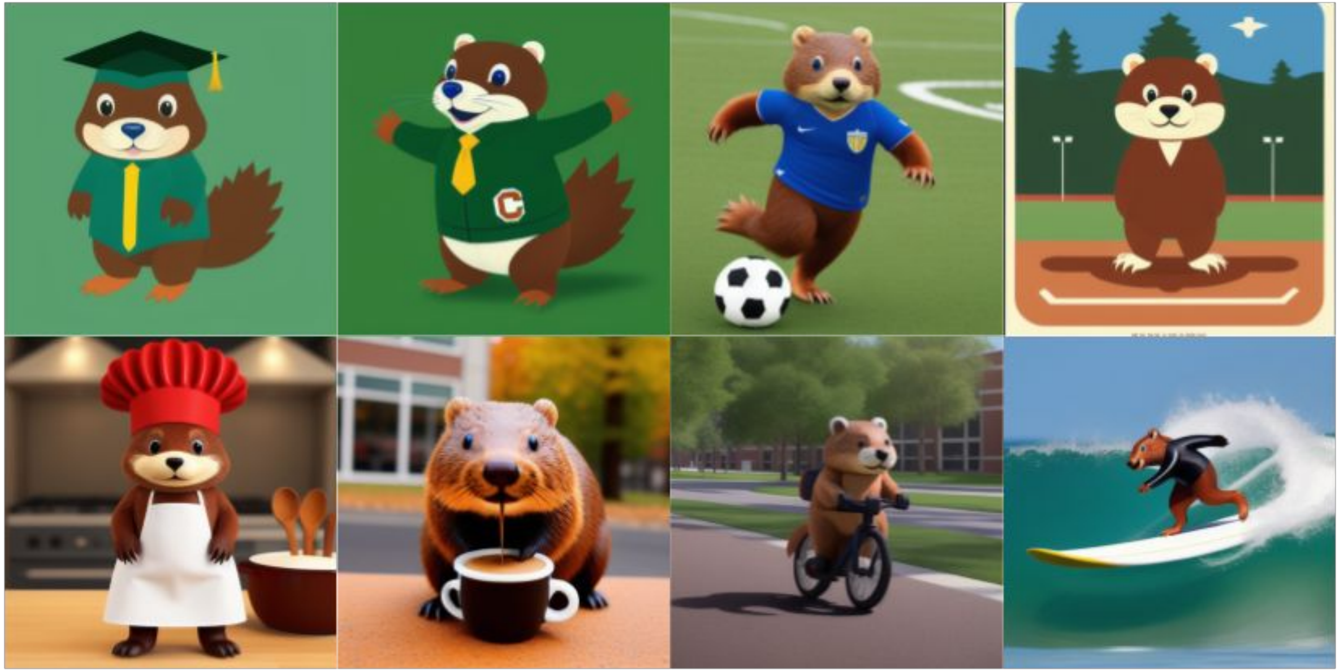
```
[ ] #Definitely using this search function
    #uses sklearn tfidf which searches for special words in the query and finds file names that have these rare words in them
    from sklearn.feature_extraction.text import TfidfVectorizer
    from sklearn.metrics.pairwise import cosine_similarity

    #file_names = [
        #'chauncey the beaver,cartoon,green,reading,reading,in park.png',
        #'chauncey the beaver,cartoon,orange and blue,in motion,playing baseball,on field.png',
        #'chauncey the beaver,cartoon,blue,like he\'s holding something,looking at computer screen,in front of computer screen.png'
    #]

    vectorizer = TfidfVectorizer()
    corpus = file_names
    tfidf_matrix = vectorizer.fit_transform(corpus)
    query_vector = vectorizer.transform([query])
    similarities = cosine_similarity(query_vector, tfidf_matrix)

    results = [(similarity, file_name) for similarity, file_name in zip(similarities[0], file_names)]
    results = sorted(results, key=lambda x: x[0], reverse=True)

    for similarity, file_name in results[:1]:
      #prints the top match to query
        print(f"{file_name}: {similarity:.4f}")
```

```
chauncey the beaver,cartoon,orange and blue,in motion,playing baseball,on field.png: 0.4600
```

Logo Outputs

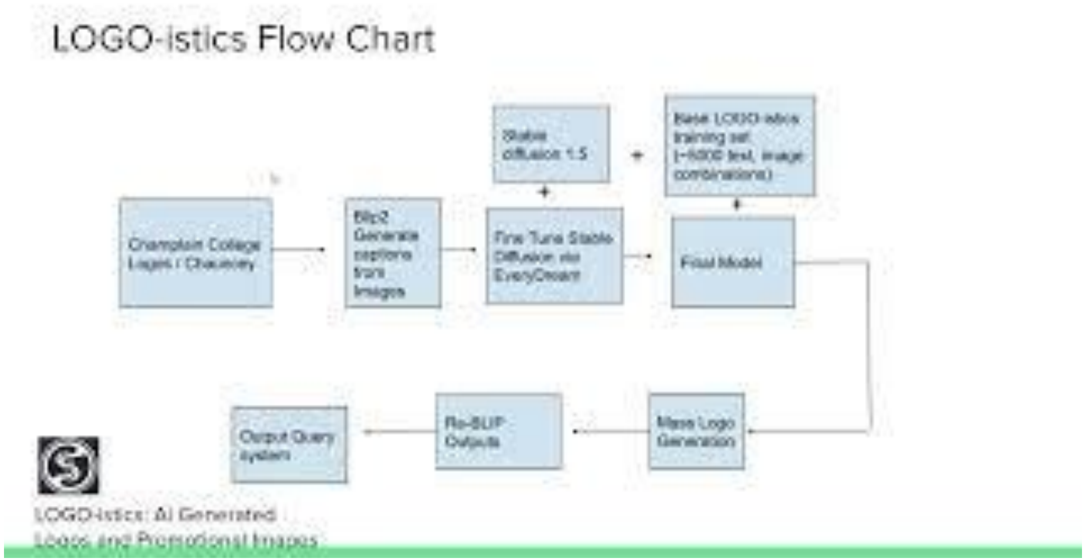


Promotional Outputs + Project Importance



LOGO-istics: Ai Generated
Logos and Promotional Images

Intro Video



LOGO-istics Flow Chart

Champlain College Logos / Chauncey

Blip2 Generate captions from images

Stable diffusion 1.5

Base LOGO-istics training set (~5000 text, image combinations)

Fine Tune Stable Diffusion via EveryDream

Final Model

Mass Logo Generation

Re-BLIP Outputs

Output Query system

LOGO-istics: AI Generated Logos and Promotional Images

Code Video