PoseFramer: The Next Iteration Of Art Reference

The First Thing That Came To Our Minds Was...

Mood Boards!


GOLD HOOPS!
CRESCENT PURSE
OVERSIZED
CROP


PERSEVERANCIA
TÚ PUEDES
HÁBITOS SALUDABLES
VISION BOARD
AMOR
LOS OBSTÁCULOS ME HACEN MÁS FUERTE
SOY VALIENTE
SOY EXITOSA
SOY CAPAZ

How Do Reference Boards Actually Help Artists?

The Character Illustration Pipeline

Ideation → **Reference** → **Finalization**

Finding Reference Is A Pain

Endless searching for the perfect pose

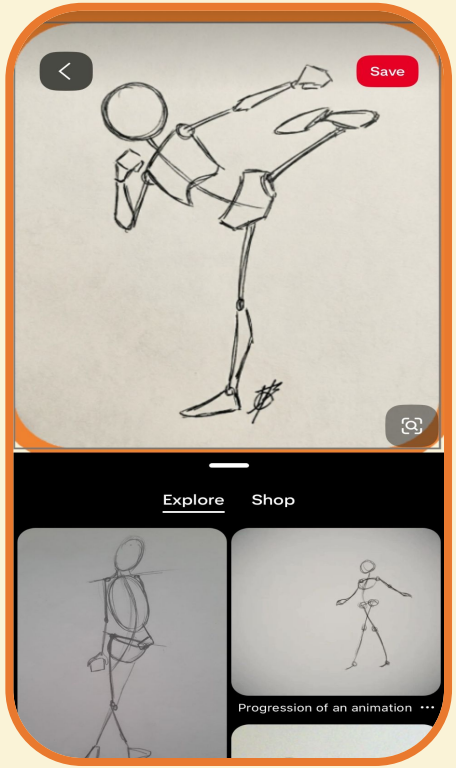
Compromising with reference not close to the initial ideation

Anatomical errors caused by poor reference material

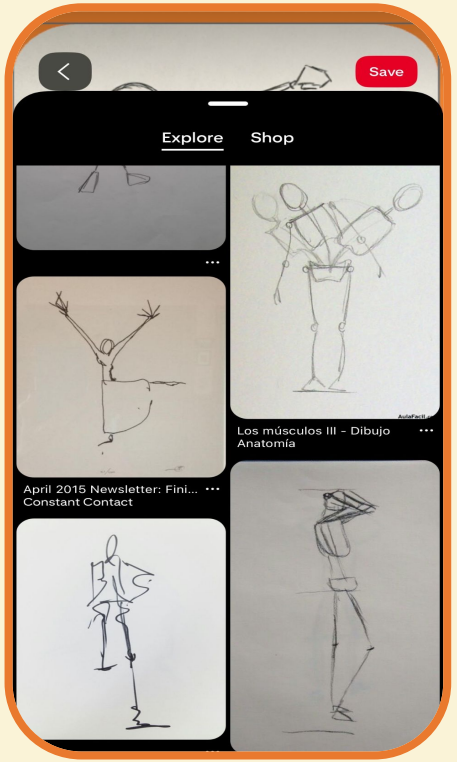
Existing Technology Is Not Sufficient

Text Search

Existing Technology Is Not Sufficient



Reverse Image Search



Our Goal?

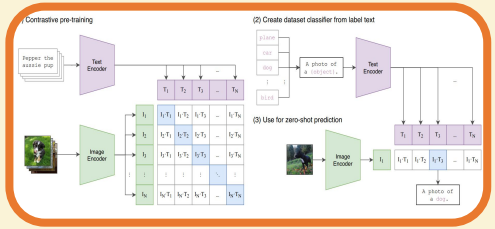
Allow Artists To Create Perfect Mosaics Without Compromising On Their Creative Vision

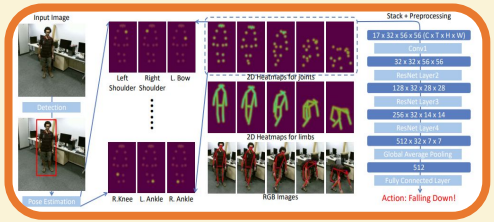
We Propose A Novel Algorithm

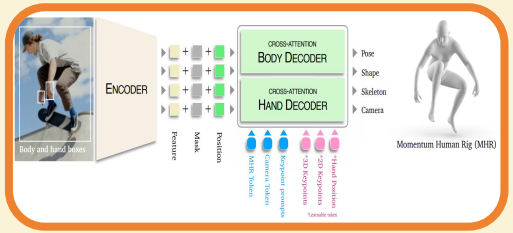
Text Search + **Pose Search**

- Clip

- Sam 3D Body
- Pose C3D





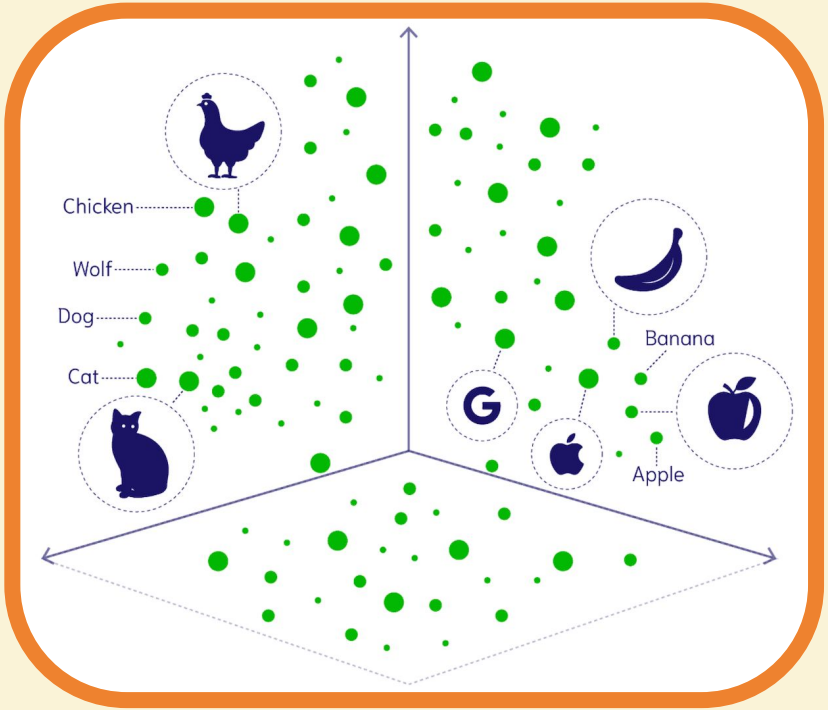


Demo

Pinterest Demo

How Does It Work?

Embedding Spaces



The Pose Embedding Pipeline

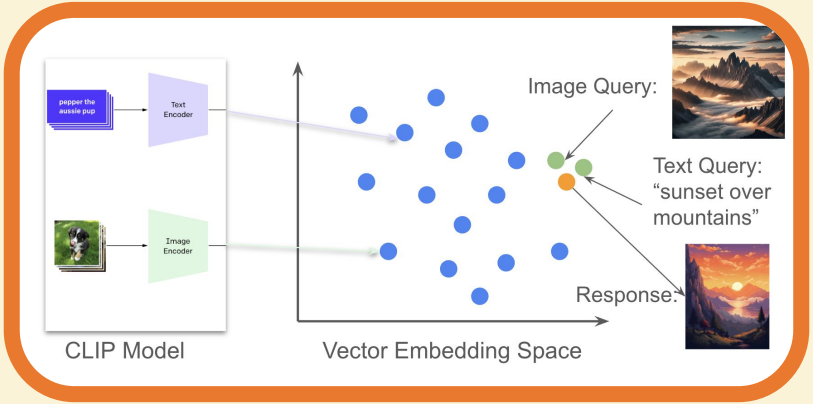
Run Sam3D Body to retrieve a 2D pose

Prune any extra joints

Retrieve an embedding vector from PoseC3D

Clip Embeddings

- Embeds text and images
- If image and text represent the same concept, the embedding is the same



Scoring

$$Sim = L * Pose\_Sim + (1 - L) * Clip\_Sim$$

$$Pose\_Sim = f(ImageA, ImageB)$$

$$Clip\_Sim = f(Description, ImageB)$$

Example Output

Image A
Joints: 70

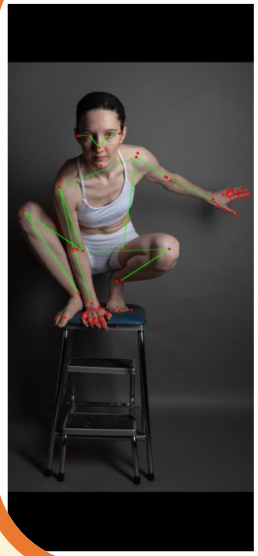


Image B
Joints: 70

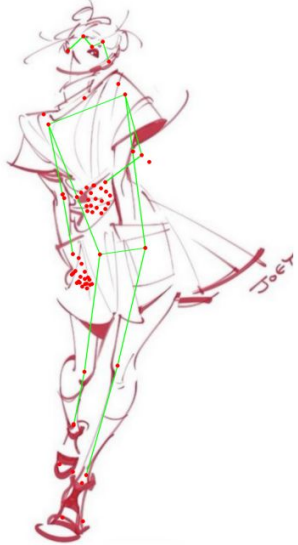
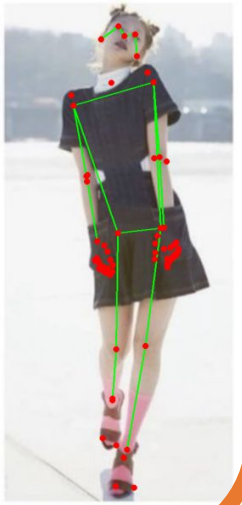


Image C
Joints: 70



Modal

- Our models are not LLMs - there's no existing API's for them
- We had to build API's for each with modal

Future Application

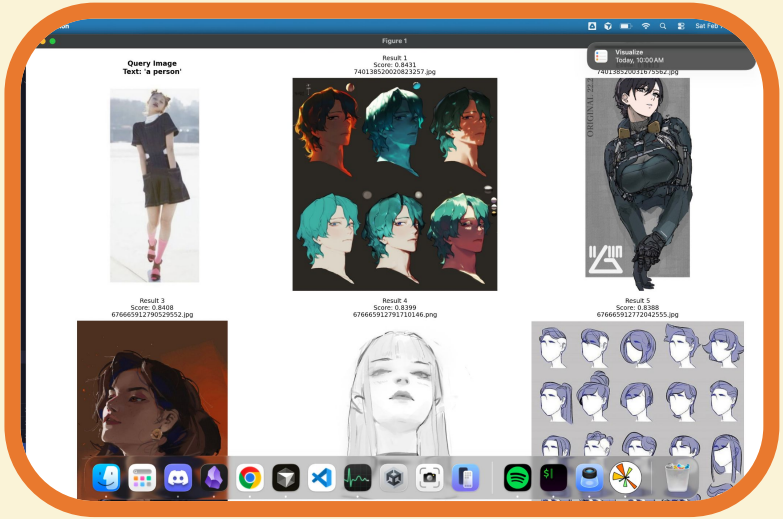
Caveats

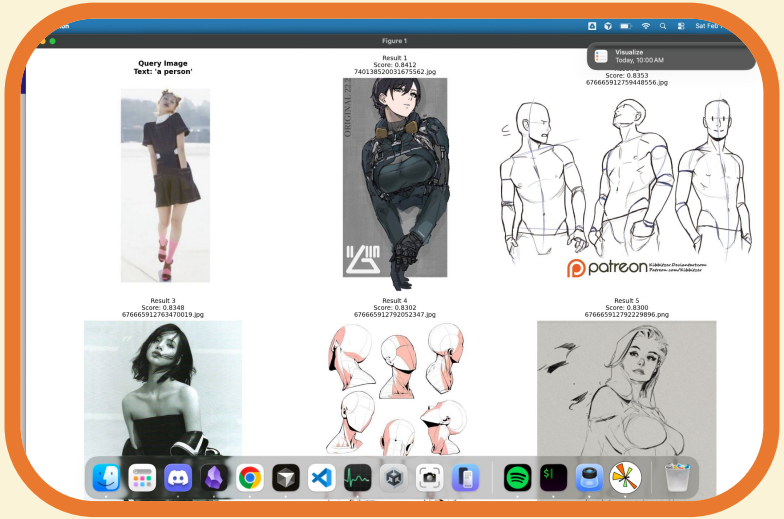
Small database of images to search from

Database is polluted with portraits

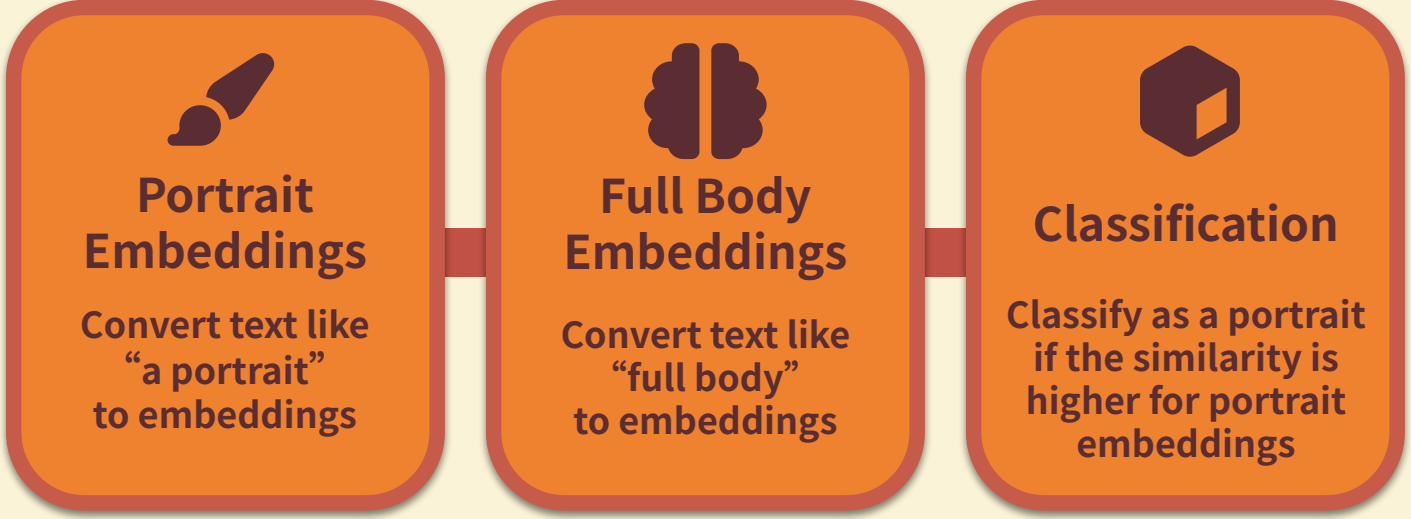
Sketches created outside of app

Portrait Filtering

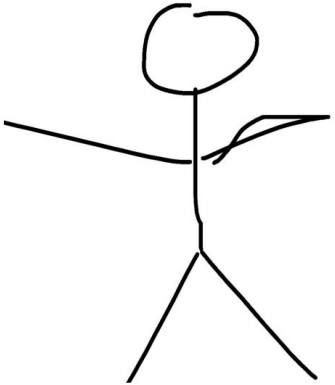




Portrait Filtering



Query Image
Text: 'a person'



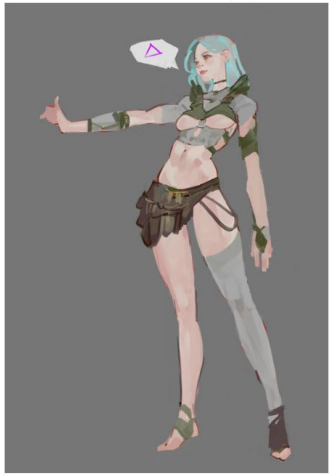
Result 1
Score: 0.8344
676665912791245166.jpg

Result 2
Score: 0.8327
676665912785446501.jpg

Result 3
Score: 0.8327
676665912791448721.jpg

Result 4
Score: 0.8290
676665912785300782.jpg

Result 5
Score: 0.8281
740138520022323079.png



THANK YOU