DOCUMENTATION FOR GLANCE ASSIGNMENT
BY- M.MEYYAPPAN

APPROACHES:
Since we are developing a fashion retrieval system we need t consider the data reliability and precision along with the retrieval speed.
Here I have used the fashionpedia dataset and they have the metadata along with them , but they are useful when we need to perform a quick retrieval and not something with precision, which is the main drawback.
To understand the color , the vibe , the surroundings only metadata based retrieval is not useful so we use Vision Language Models to create descriptions for the images which is the synthetic metadata and then use a cross – encoder to finally perform a precise accurate retrieval of the images , using the natural language queries, but this approach comes with a tradeoff which is speed , but considering that we have a ample VRAM in a Gpu we can use this method.
GITHUB LINK : https://github.com/mm0177/GLANCE-ML/tree/main

Version -1 (Approach 1)

So here with the first approach we have used a Decomposed Semantic Retrieval Flow , which separates it from the standard Clip Model Usage.

We have used Microsoft-Florence-2 Model to create a detailed caption for the images, it defines each of the elements like the background , the material texture etc.

Then we use the Vit-L/14 which encodes the image into 5 distinct vectors into a local Qdrant Database which includes (Global , Clothing, Color, Scene, Style).

Then we have the Regrex Based Query Parser , which analyses the user's text , where in it maps the query using Weights. This ensures that the search engine understands which part of the user's query is most important.

Then finally we have the Cross -encoder (ms-marco-MiniLM-L-6-v2) ,where in after the Clip Model retrieves the top 50 images , the the cross encoder takes the user query and the AI generated metadata from Florence-2 and compare them directly in a single neural pass

This is how the first approach is made.

Lets look at the output

Query 2: 'Professional business attire inside a modern office'

#1 (Score: 4.704)

pocket, lapel, coat +4 more
office

#2 (Score: 4.095)

shoe, belt, skirt +3 more
office

#3 (Score: 4.071)

shoe, jacket, bag, wallet +6 more
office

#4 (Score: 3.986)

shoe, pants, bag, wallet +3 more
office

#5 (Score: 3.534)

zipper, pocket, lapel +7 more
office

Query 3: 'Someone wearing a blue shirt sitting on a park bench'

#1 (Score: 1.790)

top, t-shirt, sweatshirt, neckline
park

#2 (Score: 1.715)

shirt, blouse, pocket, shoe +4 more
park

#3 (Score: 1.531)

shirt, blouse, dress, hat +2 more
park

#4 (Score: 1.454)

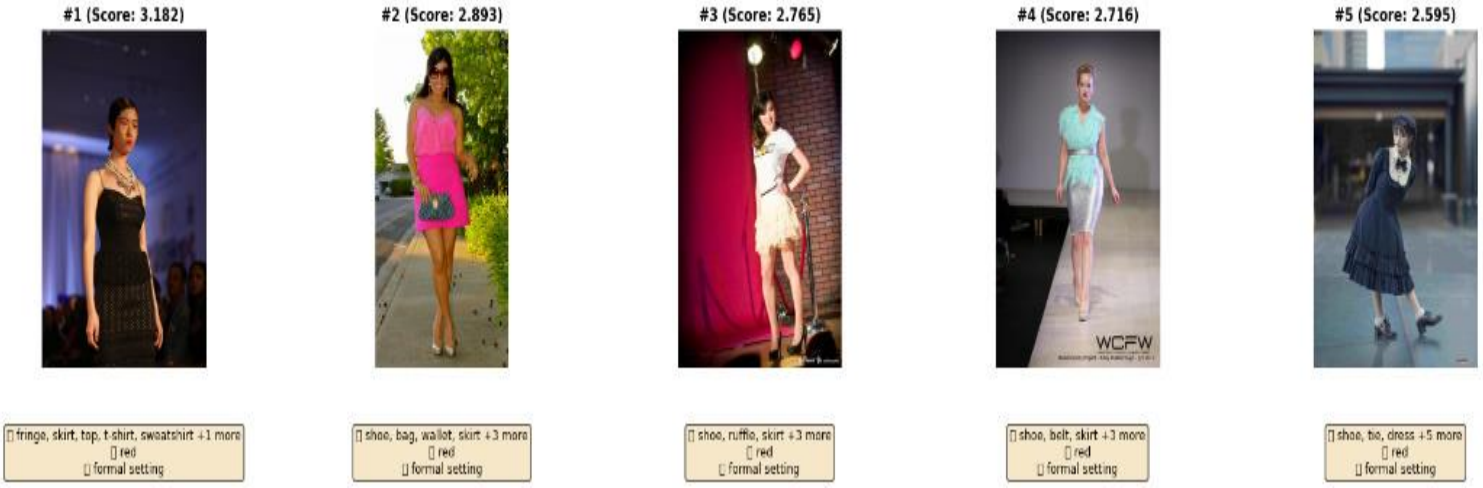
neckline, pocket, pants +2 more
park

#5 (Score: 1.374)

skirt, shirt, blouse
park

Query 4: 'Casual weekend outfit for a city walk'

Query 5: 'A red tie and a white shirt in a formal setting'



I have put only the images of 5 queries in the document , to check the output of all the 10 queries Kindly view the Notebook in Github.

Link: https://github.com/mm0177/GLANCE-ML/blob/main/VERSION-1.ipynb

Version 2 (Approach -2)

In the version 2 , the captions which we get from the Florence-2 are vectorized and saved in Qdrant DB (compared to version 1 where the captions were used just as metadata).

For the CLIP model (ViT-L/14) , it performs a Zero Shot Color Detection , where in it uses prompt ensembling (which means “ is it crimson” , “is it scarlet”) , and verifies and corrects the dataset tags to ensure that even if an image is untagged it is correctly indexed.

For the query Parser , I have made it intent aware , where in it uses Synonym Mapping and Adaptive Weighting. It analyses the complexity of query and if the query is long , then it shifts the search power from simple tags to Florence-2 caption.

For the Cross-Encoder we are using an L-12 , to perform better.

I have also introduced a new function which is search by image which the CLIP Global and Clothing vectors to find "Visual Twins." It ignores the text pathway entirely to focus on the silhouette, cut, and "vibe" of an uploaded image.

Query 1: 'A person in a bright yellow raincoat'

#1 (Score: -4.677)

collar, glasses, top, t-shirt, sweatshirt +7
yellow
street

#2 (Score: -4.698)

glasses, hat, top, t-shirt, sweatshirt +7
yellow
street

#3 (Score: -5.813)

tights, stockings, dress, bag, wallet +4
yellow, orange
street

#4 (Score: -6.003)

neckline, dress
yellow
outdoor

#5 (Score: -6.037)

dress, coat, sleeve +3
yellow, orange, purple
studio

Query 2: 'Professional business attire inside a modern office'

#1 (Score: 5.315)

collar, shirt, blouse, coat +4
office

#2 (Score: 4.730)

jacket, glasses, vest +7
office

#3 (Score: 4.724)

collar, top, t-shirt, sweatshirt, watch +4
office

#4 (Score: 4.491)

top, t-shirt, sweatshirt, sleeve, shoe +3
office

#5 (Score: 4.438)

jacket, collar, shirt, blouse +5
office

Query 3: 'Someone wearing a blue shirt sitting on a park bench'

#1 (Score: 0.240)

glasses, top, t-shirt, sweatshirt, neckline +5
grey, blue, purple
park

#2 (Score: 0.237)

collar, glasses, shirt, blouse +7
blue
park

#3 (Score: -0.310)

top, t-shirt, sweatshirt, sleeve, collar
blue
park

#4 (Score: -0.645)

dress, hat, top, t-shirt, sweatshirt +3
blue, purple
park

#5 (Score: -2.245)

top, t-shirt, sweatshirt, neckline, shorts +1
tan
park

Query 4: 'Casual weekend outfit for a city walk'

#1 (Score: 1.971)

glasses, top, t-shirt, sweatshirt, bag, wallet +5
city

#2 (Score: 1.805)

jacket, scarf, bag, wallet +5
city

#3 (Score: 1.320)

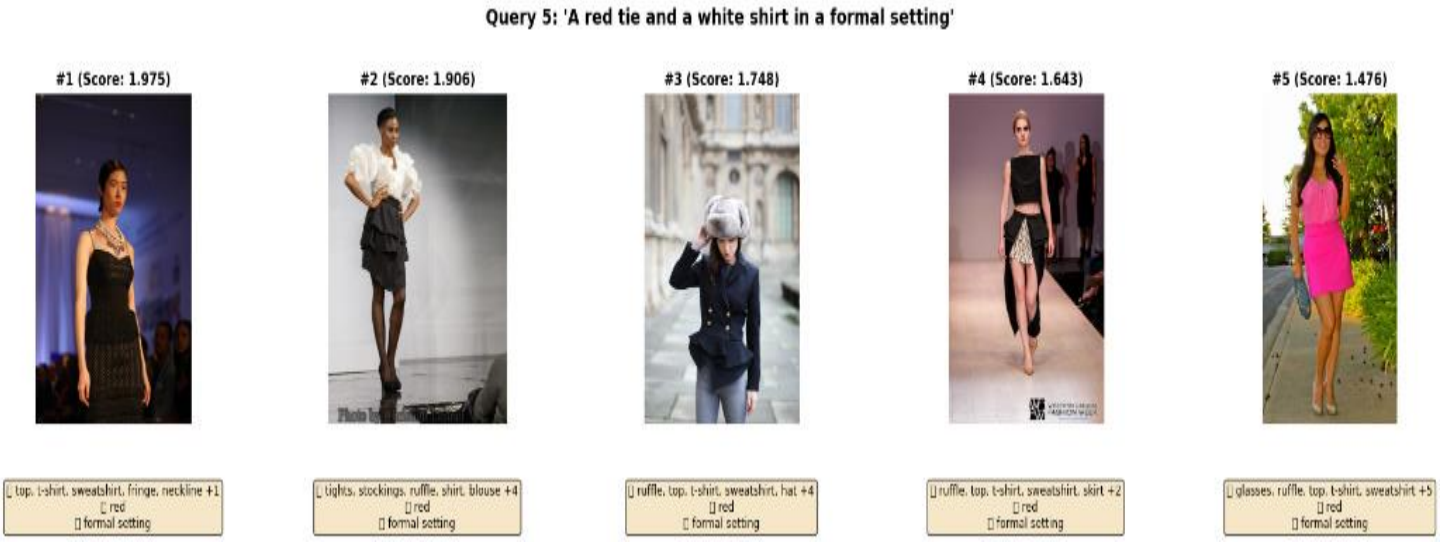
top, t-shirt, sweatshirt, sleeve, shoe +2
city

#4 (Score: 1.221)

collar, glasses, top, t-shirt, sweatshirt +8
city

#5 (Score: 0.479)

tights, stockings, rivet, top, t-shirt, sweatshirt +7
city



To check all the query outputs kindly view the github notebook
Link : https://github.com/mm0177/GLANCE-ML/blob/main/VERSION-2.ipynb

Version -3(Apporach-3)

In Version -3 we introduce Blip to work along side Florence-2 , it gives a narrative flow , which helps in getting better details of the clothing and the environment.

Now we have 7 distinct vectors per image to get a better understanding of the image.

For the color detection , in version 2 it was averaged color names (synonyms) into a single hardcoded template, but now For every color (e.g., "Crimson"), the system now generates 5+ different contextual sentences (e.g., "a photo of crimson fabric," "close-up of crimson garment," "crimson colored clothes").

Also the intent of the query is understood so like if there is more focus on a color then the search weight shifts to Florence and Clothing Vectors, but if its something more about the environment then it shifts to BLIP and Style Vectors.

Now the Reranker uses the Dataset Tags , Florence Technical Captions , Blip Narative Captions and Clip Detected colors.
Link To Notebook: https://github.com/mm0177/GLANCE-ML/blob/main/VERSION-3.ipynb

Query 3: 'Someone wearing a blue shirt sitting on a park bench'

#1 (Score: 6.918)

Clothing: dress, scarf, watch +4
Colors: green
Scene: casual setting

#2 (Score: 6.833)

Clothing: sleeve, top, t-shirt, sweatshirt, shoe +5
Colors: white, gray, purple, black +1
Scene: park

#3 (Score: 6.818)

Clothing: sock, sleeve, top, t-shirt, sweatshirt +4
Colors: red, gray, purple, black +1
Scene: park

#4 (Score: 6.462)

Clothing: top, t-shirt, sweatshirt, shoe, neckline +2
Colors: red, tan
Scene: park

#5 (Score: 6.462)

Clothing: sleeve, top, t-shirt, sweatshirt, neckline +3
Colors: white, brown, gray, black +1
Scene: beach

Query 4: 'Casual weekend outfit for a city walk'

#1 (Score: 2.790)

Clothing: lapel, pants, sleeve +6
Colors: brown, gray, white
Scene: casual setting

#2 (Score: 2.577)

Clothing: pants, sleeve, top, t-shirt, sweatshirt +7
Colors: tan, white, brown, gray +1
Scene: street

#3 (Score: 2.446)

Clothing: pants, sleeve, collar +6
Colors: purple, tan
Scene: street

#4 (Score: 2.383)

Clothing: sleeve, top, t-shirt, sweatshirt, shoe +5
Colors: black, tan, white, gray
Scene: street

#5 (Score: 2.326)

Clothing: pants, sleeve, top, t-shirt, sweatshirt +7
Colors: brown, gray, purple, black +1
Scene: casual setting

Query 5: 'A red tie and a white shirt in a formal setting'

#1 (Score: 6.671)

Clothing: lapel, tie, shirt, blouse +6
Colors: white, brown, red, gray +1
Scene: formal setting

#2 (Score: 6.670)

Clothing: watch, pants, belt +6
Colors: white, orange, brown, red +1
Scene: urban

#3 (Score: 5.954)

Clothing: lapel, pants, tie +7
Colors: orange, brown, red, gray +1
Scene: casual setting

#4 (Score: 5.751)

Clothing: collar, sleeve, shirt, blouse +3
Colors: white, orange, brown, red +1
Scene: formal setting

#5 (Score: 5.503)

Clothing: lapel, pants, tie +5
Colors: brown, black, gray, white
Scene: formal setting

FUTURE WORK AND IMPORVEMENTS

1) To incorporate location and weather awareness, you could enrich your system with **geo-temporal contextual embeddings** by training a specialized adapter that maps location-weather pairs (e.g., "Tokyo summer", "NYC winter") to appropriate fashion attributes. This could be achieved by collecting a dataset of fashion images tagged with location metadata and weather conditions, then fine-tuning a small auxiliary network that modulates your existing CLIP embeddings based on contextual factors

2) Currently the version 1 uses 10k images , the version 2 uses 15k images and the version 3 uses 20k images. To imporve the performace we can increase it to 35k images and also create a fine tuned model for the color expertise so that the color matching can be improved.

3) Also we can use user feedback so that using the query the model can understand the user preference over -time.