IR ASSIGNMENT-2

Mahisha Ramesh- MT23121

TASK-1

1) Image Feature Extraction Using ResNet50:-

a. Image Pre-processing:-

- Resizing: All images were resized to a standard size suitable for input to the ResNet50 model, typically 224x224 pixels.
- Geometrical Orientation: Images were adjusted to ensure correct orientation for processing.
- Augmentation: Random flips (horizontal and vertical) were applied to augment the dataset and improve model generalization.
- Contrast and Brightness Adjustment: Alterations were made to enhance visual features in the images.

```
# Function to download and preprocess image with random augmentation
def preprocess_image(url):
    response = requests.get(url)
    img = Image.open(BytesIO(response.content))

    # Resize
    img = img.resize((224, 224))

    # Random flips
    if np.random.rand() < 0.5:
        img = img.transpose(Image.FLIP_LEFT_RIGHT)
    if np.random.rand() < 0.5:
        img = img.transpose(Image.FLIP_TOP_BOTTOM)

    # Brightness adjustment
    enhancer = ImageEnhance.Brightness(img)
    img = enhancer.enhance(np.random.uniform(0.5, 1.5))

    # Exposure adjustment
    enhancer = ImageEnhance.Contrast(img)
    img = enhancer.enhance(np.random.uniform(0.5, 1.5))

    # Convert to numpy array
    img = np.array(img)

    # Preprocess for ResNet50 model
    img = preprocess_input(img)

    return img
```

b. Feature Extraction with ResNet50:

- We utilized ResNet50, a pre-trained CNN model, which was trained on the ImageNet dataset.
- Each pre-processed image was passed through ResNet50 to extract high-level features.
- Features were extracted from a specific layer, typically before the classification layers, to capture semantic information.

Task -2

Text Feature Extraction Using TF-IDF:

a. Text Pre-processing:

- Lower-Casing: All text was converted to lowercase to ensure uniformity.
- Tokenization: Text was split into individual tokens or words for further analysis.
- Punctuation Removal: Punctuation marks were removed from the text.
- Stop Word Removal: Common stop words were excluded to focus on content-bearing words.
- Stemming and Lemmatization: Words were reduced to their base or root form to standardize variants.

```
def preprocess_text(text):

    # Lowercasing
    text = text.lower()

    # Tokenization
    tokens = word_tokenize(text)

    # Removing punctuations
    tokens = [token for token in tokens if token not in string.punctuation]

    # Stop word removal
    stop_words = set(stopwords.words('english'))
    tokens = [token for token in tokens if token not in stop_words]

    # Lemmatization using spaCy
    doc = nlp(" ".join(tokens))
    tokens = [token.lemma_ for token in doc]

    # Joining tokens back to text
    processed_text = ' '.join(tokens)

    return processed_text
```

b. Calculation of TF-IDF Scores:

- Term Frequency (TF): Frequency of each term in a document divided by the total number of terms in the document.
- Inverse Document Frequency (IDF): Logarithm of the total number of documents divided by the number of documents containing the term.
- TF-IDF Score: Product of TF and IDF scores for each term in each document.


```
TF-IDF Matrix:
[{'love': 0.13095286797705646, 'vintage': 0.5173800274081972, 'spring': 0.6945981537121895, 'strat': 0.1999994321942258, 'good': 0.08983916897305623,
'tension': 0.3227840915144069, 'great': 0.07533781605573306, 'stability': 0.35515127923318696, 'float': 0.3339210304919513, 'bridge': 0.22308451464208
14, 'want': 0.13858897075920396, 'way': 0.17083490958126993, 'go': 0.12055258474528437}
 {'work': 0.06045305669636706, 'great': 0.047086135034833165, 'guitar': 0.05017396720790491, 'bench': 0.2415530340726475, 'mat': 0.2415530340726475,
'rugge': 0.23395630253956604, 'enough': 0.10331071020918482, 'abuse': 0.22196954952074183, 'take': 0.18588008313365093, 'care': 0.339542185000203, 'ma
ke': 0.058553993842353755, 'organization': 0.2508506820440729, 'workspace': 0.2508506820440729, 'much': 0.08130790061749744, 'easy': 0.079055099731235
96, 'screw': 0.1077678818571918, 'will': 0.10182775416323887, 'not': 0.04084259469666456, 'roll': 0.19577752204480958, 'around': 0.10612076357566533,
'color': 0.11430298483606555, 'good': 0.05614948060816014}
 {'work': 0.06045305669636706, 'great': 0.047086135034833165, 'guitar': 0.05017396720790491, 'bench': 0.2415530340726475, 'mat': 0.2415530340726475,
'rugge': 0.23395630253956604, 'enough': 0.10331071020918482, 'abuse': 0.22196954952074183, 'take': 0.18588008313365093, 'care': 0.339542185000203, 'ma
ke': 0.058553993842353755, 'organization': 0.2508506820440729, 'workspace': 0.2508506820440729, 'much': 0.08130790061749744, 'easy': 0.079055099731235
96, 'screw': 0.1077678818571918, 'will': 0.10182775416323887, 'not': 0.04084259469666456, 'roll': 0.19577752204480958, 'around': 0.10612076357566533,
'color': 0.11430298483606555, 'good': 0.05614948060816014}
 ...
 {'really': 0.03325107797204334, 'like': 0.024486242084920232, 'simplicity': 0.09101334301870545, 'bridge': 0.1115422573210407, 'adjust': 0.0549306144
3340549, 'easy': 0.03162203989249439, 'string': 0.14504014088820358, 'height': 0.1313658275792638, 'length': 0.0607585102414013, 'come': 0.03513988939
382673, 'mount': 0.09338468480631829, 'screw': 0.12932145822863017, 'even': 0.03617114612246152, 'small': 0.03680356164853024, 'allen': 0.096621213629
059, 'wrench': 0.096621213629059, 'make': 0.04684319507388301, 'adjustment': 0.06196385460106173, 'provide': 0.05635598399070819, 'phillip': 0.0935825
2101582641, 'head': 0.05665629241575282, 'one': 0.021810607466856667, 'thing': 0.03824538236368495, 'miss': 0.07270313820757028, 'order': 0.0465265035
5560635, 'fact': 0.07047761499716157, '-': 0.08350692163492242, 'thru': 0.17364953842804737, 'guitar': 0.020069586883161964, 'body': 0.056355983990708
19, 'placement': 0.08682476921402368, 'dead': 0.07622495643535707, 'end': 0.04702905419178448, 'stop': 0.06790843700004061, 'flair': 0.091013343018705
45, 'plate': 0.06467250342602465, 'include': 0.0423196046620343, 'picture': 0.03906859495251123, 'may': 0.053366968513111235, 'help': 0.06037535160432
298, 'other': 0.05759111685993688, 'aware': 0.07723536679896431, 'application': 0.08069602287860173, 'biggie': 0.10513497402515884, 'tho': 0.093582521
01582641, 'use': 0.017684572822484076, 'another': 0.0440756132563314, 'project': 0.0719278046136554}
 {'great': 0.09417227006966633, 'product': 0.1633311132092775, 'warranty': 0.5645423715351845, 'usa': 0.18379076902598537, 'purchase': 0.1932595329121
0468, 'amazon': 0.1255108989793834, 'hope': 0.15815803406389234, 'never': 0.12602850397932328, 'use': 0.044211432056210184, 'amazing': 0.1334174212827
781, 'accord': 0.2797318145674039, 'ride': 0.3973044500468651, 'see': 0.0785520755131414, 'screenshot': 0.2797318145674039, 'microphone': 0.2863682347
187932, 'amazon.com': 0.2797318145674039, 'pity': 0.2797318145674039}
 {'product': 0.17817939622830276, 'good': 0.12250795769053123, 'use': 0.0964613063044586, 'professional': 0.39656257204129247, 'mike': 0.4640114215621
45, 'date': 0.5104501146317805, '..': 0.3214099744790681, 'fast': 0.350123878971657, 'shipping': 0.42128381890344174, 'worth': 0.28454951010559865, 'b
uy': 0.15572532648863097}]
```


TASK-3

Image based retrieval

To conduct image-based retrieval with associated text and cosine similarity scores, the methodology involves extracting features from both images and text, calculating image similarity using cosine similarity, selecting the top matches, and determining textual similarity using the same metric. Specifically, for an input image-review pair (I1, R1), the most relevant images (Ia, Ib, Ic) are identified based on their visual similarity to I1. Corresponding review texts (Ra, Rb, Rc) are retrieved for these images, and cosine similarity scores between R1 and each matched review are computed. The results are then presented in the specified format, comprising the matched images along with their corresponding reviews and cosine similarity scores, providing users with insights into both visual and textual relevance.

Results:-

Image:

https://images-na.ssl-images-amazon.com/images/I/71bztfqdg+L._SY88.jpg

1) **Cosine similarity of top 3 relevant images**

Image Number: 887

Image: https://images-na.ssl-images-amazon.com/images/I/5134EWdp6lL._SY88.jpg

Text: love guitar honestly never hold squire brand strat compare everything ' ve hold play hear past guitar really give good bang buck especially be like pretty much poor dirt think guitar cheap would probably want get small upgrade ' m pretty content get first 10 minute play recommend guitar pretty much anyone want strat incredible quality price

Cosine Similarity of Image: 0.7026417

Image Number: 1099

Image: https://images-na.ssl-images-amazon.com/images/I/61Wqw4GwL8L._SY88.jpg

Text: ' ve bough work good sure else say cheap - look hold violin bow wall way

Cosine Similarity of Image: 0.83370674

Image Number: 1244

Image: https://images-na.ssl-images-amazon.com/images/I/71N0t6HU37L._SY88.jpg

Text: man word say catalina maple drum set place head first hit sound like cannon perfect sound make want play day make want start dance eye set product church product know would next purchase market new drum head hard hitter go evans emad2 disappoint

Cosine Similarity of Image: 0.8736638

Cosine similarity of texts corresponding to top 3 relevant images

```
Enter three indices separated by spaces:  887 1099 1244
Top 86: Cosine Similarity between user dictionary and vector 887: 0.07959818016311464
Top 148: Cosine Similarity between user dictionary and vector 1099: 0.05487406744368 74
Top 1487: Cosine Similarity between user dictionary and vector 1244: 0.0
```

+ Code + Markdown

Composite scores

```
Enter the cosine similarities for pair 1 (image, review) separated by space:  0.7026417 0.07959
Enter the cosine similarities for pair 2 (image, review) separated by space:  0.83370 0.05487
Enter the cosine similarities for pair 3 (image, review) separated by space:  0.87366 0
Rank 1: image3_review3 - Composite Score: 0.6115619999999999
Rank 2: image2_review2 - Composite Score: 0.6000509999999999
Rank 3: image1_review1 - Composite Score: 0.51572619
```

TASK -4

TEXT BASED RETRIEVAL

We employ a text-based retrieval system that ranks documents based on their textual content. Additionally, we calculate the Cosine Similarity scores for pairs of images to measure their similarity. For each query, we retrieve the top-ranked documents based on text (Ia, Ib, Ic) and their corresponding ranking (Ra, Rb, Rc). Simultaneously, we retrieve the top-ranked images based on similarity scores and their associated ranking.

Results:-

Cosine similarity of top 3 relevant text with the given review

Review: I have been using Fender locking tuners for about five years on various strats and teles. Definitely helps with tuning stability and way faster to restring if there is a break.

Cosine similarity of texts

Top 1: Cosine Similarity between user dictionary and vector 1245: 0.8306238893131533

Full row from CSV: ['1245', 'https://images-na.ssl-images-amazon.com/images/I/71bztfqdg+L._SY88.jpg', 'use fender lock tuner five year various strat tele definitely help tune stability way fast restring break']

Top 2: Cosine Similarity between user dictionary and vector 1: 0.2811710011960442

Full row from CSV: ['1', 'https://images-na.ssl-images-amazon.com/images/I/81q5+IxFVUL._SY88.jpg', 'love vintage spring vintage strat good tension great stability float bridge want spring way go']

Top 3: Cosine Similarity between user dictionary and vector 621: 0.2418052370248125

Full row from CSV: ['621', 'https://images-na.ssl-images-amazon.com/images/I/61Np-qH9ZVL._SY88.jpg', 'fit tele build perfectly look great']

Cosine similarity of images corresponding to top 3 relevant texts

```
Number of keys present in the dataset: 1648
Enter index 1 (between 0 and 1647):  1245
Enter index 2 (between 0 and 1647):  1
Enter index 3 (between 0 and 1647):  621
Index: 1245, Cosine Similarity of image: 0.4557837247848511
Index: 1, Cosine Similarity of image: 0.505005419254303
Index: 621, Cosine Similarity of image: 0.398740828037261 96
```

Composite scores

```
Enter the cosine similarities for pair 1 (image, review) separated by space:  0.45578 0.83062
Enter the cosine similarities for pair 2 (image, review) separated by space:  0.50500 0.28117
Enter the cosine similarities for pair 3 (image, review) separated by space:  0.39874 0.24180
Rank 1: image1_review1 - Composite Score: 0.5145600000000001
Rank 2: image2_review2 - Composite Score: 0.314468
Rank 3: image3_review3 - Composite Score: 0.256216
```

Conclusion

TEXT RETRIEVAL IS BETTER THAN IMAGE RETRIEVAL SYSTEM

- While cosine similarity measures the similarity between feature vectors representing images, it may not capture the semantic relevance or context of the images accurately.
- Even though two images may have similar visual features, they could represent different concepts or contexts, leading to a mismatch between the similarity score and user expectations.
- In some cases, text-based retrieval systems may excel at capturing the exact review or textual content related to the query. However, the corresponding images retrieved based on similarity scores may not align perfectly with the textual content due to differences in representational modalities

- . Even if the cosine similarity between images is high, the relevance of images to the textual context may not be accurately captured.