TEXT+IMAGE MULTIMODAL SEARCH USING MOBILENET

D.Hemanth Kumar[1], P.Naveen[2], P.Pavan Kumar[3]
Shaik Rafi[4]

[1,2,3.]Student, Department of CSE, Narasaraopeta Engineering Collage, Narasaraopeta, Guntur(D.T), Andhra Pradesh, India.

[4] Professor, Department of CSE, Narasaraopeta Engineering Collage, Narasaraopeta, Guntur(D.T), Andhra Pradesh, India.

hemanthkumardarsi4@gmail.com [1], pittanaveen3315@gmail.com [2], paramkusampavankumar1@gmail.com[3], shaikrafinrt@gmail.com[4]

ABSTRACT:

Multi-modal search is a task of retrieving relevant results from a database using multiple modalities such as text and images. The goal of the multi-modal search is to provide more accurate and comprehensive search results by integrating different types of data. This method is frequently employed across a number of industries, including e-commerce, healthcare, social media, and entertainment. The Multi-modal search requires the use of various techniques such as feature extraction, similarity measures, and machine learning algorithms. The Multi-modal search has grown in importance as a study topic in the fields of information retrieval and computer vision as a result of the expansion of multi-modal data availability.

1.INTRODUCTION:

The Multimodal search is a type of search technique where the system retrieves results based on the user's query using different modalities, such as text, image, audio, and video. It combines different modalities to achieve more accurate and relevant search results. In a typical multimodal search system, the input query may consist of text, image. The system extracts relevant features from each modality and combines them to form a joint feature representation. The joint representation is then used to retrieving the relevant results based on the user's query. For example, in an e-commerce website, the user may enter a query for a product, and the system may retrieve the results based on the text description of the product, as well as the image of the product.

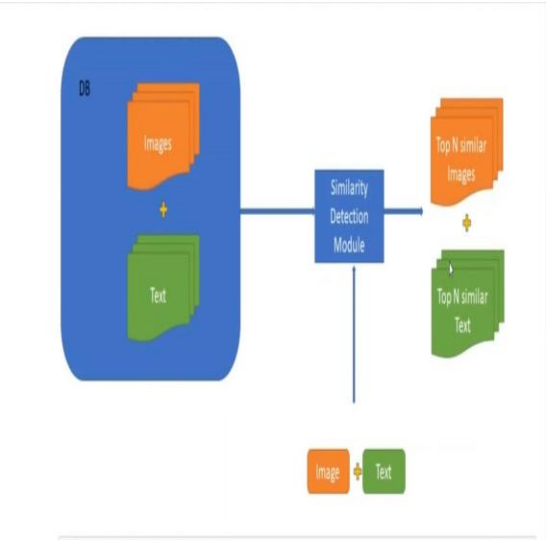


Fig 2(a) Architecture of the MultiModel Search

2.LITERATURE SURVEY:

Written by Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell, "Learning Multi-modal Similarity" is a research paper. This paper proposes a method for learning a similarity metric that can compare both textual and visual representations of objects. The method uses a combination of triplet loss and cross-modal matching to learn a joint embedding space for the two modalities.

"Multi-Modal Search with Image and Text Queries" by Yushi Jing and Katja Hofmann. This paper proposes a multi-modal search system that allows users to search for images using text queries and vice versa. The system uses a combination of deep learning and information retrieval techniques to perform the search.

"Multi-Modal Deep Learning for Image and Text Recognition" by Hyunjung Shin and Dongsuk Yook. This study suggests a deep learning paradigm for text and picture recognition that mixes recurrent and convolutional neural networks.

3. PROPOSED SYSTEM:

3.1 The Basic Mechanism:

Identify the Text and Image Datasets: The first step in building a multi-model search system for text and images is to identify the datasets that will be included in the search. These could include databases of text documents, image repositories, or other sources of information.

Data Preprocessing: Once the datasets have been identified, the data needs to be preprocessed. This involves cleaning and standardizing the data so that it can be easily searched.

Text Embedding: The text data needs to be Embedded so that it can be searched quickly and efficiently. This involves creating the vectors for the text data that can be used to search for specific terms or phrases.

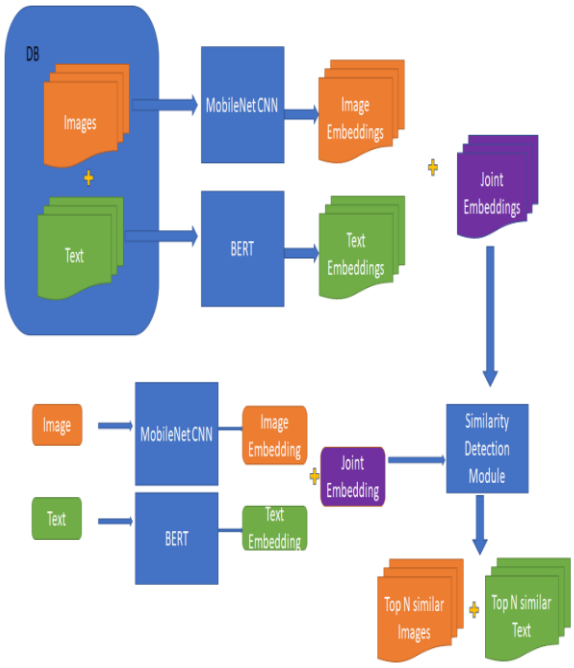
Image Embedding: The image data needs to be Embedded as well. In order to do this, features from the photos, such as colours, forms, and textures, must be extracted using computer vision algorithms. Afterwards, you may use these characteristics to look for related photographs.

Query Processing: When a user enters a search query, the system will process the query and search across both the text and image datasets. This involves using the text index to identify relevant text results and using the image features to identify

relevant image results. The results can then be ranked based on relevance.

Machine Learning: To improve the accuracy and relevance of search results, the machine learning techniques can be applied. To enhance text search results, for instance, natural language processing (NLP) models might be applied, while image recognition models can be used to improve image search results. Large volumes of data can be used to train these models to find patterns and relationships in the data that can be utilised to enhance search results.

3.2 Working Mechanism:



Here the Database contains both images and texts. Now we are using MobileNet CNN for the classification of Images and BERT for the Text Representation.

3.2.1 MobileNet CNN for Image Classification:

Convolutional neural network (CNN) architecture called MobileNet is made to be effective for embedded and mobile devices. This is accomplished by combining pointwise convolutions with depthwise separable convolutions.

Using depthwise separable convolutions, a standard convolution is split into two independent processes: a depthwise convolution, which applies a single filter to each input channel separately, and a pointwise convolution, which combines the output of the depthwise convolution using a 1x1 convolution.

As opposed to utilising a single filter for all input channels, depthwise convolution applies a distinct filter to each input channel, reducing the number of parameters in the network. This can significantly lower the convolution's computational cost.

The pointwise convolution essentially performs a linear transformation on the result of the depthwise convolution by combining it with a 1x1 convolution. As a result, the network can learn more intricate correlations between the input and output, which increases its expressive capacity.

Overall, to lower the number of parameters in the network and increase its efficiency for mobile and embedded devices, the MobileNet architecture combines depthwise separable convolutions with pointwise convolutions.

MobileNet can still achieve high accuracy on a variety of image recognition tasks despite its effectiveness.

A typical convolutional neural network (CNN) for image classification consists of several layers, including:

Input layer: The input image, which is commonly shown as a matrix of pixel values, is fed into this layer.

Convolutional layer: This layer applies to the input image a number of filters, each of which recognises a particular feature, such as edges or corners. The result of this layer is a collection of feature maps that show where each filter was activated in the image.

Activation layer: This layer provides non-linearity into the model and enables it to learn more complicated features by applying a non-linear activation function to the convolutional layer's output.

Pooling layer: This layer decreases the spatial dimensionality of the feature maps by Down sampling them, which helps to reduce the number of parameters in the model and prevent overfitting.

Dropout layer: During training, this layer randomly removes a portion of the neurons from the model, which aids in avoiding overfitting and enhancing generalisation.

Fully connected layer: Every neuron in the layer before and every neuron in the layer after are connected in this layer, enabling the model to learn intricate non-linear correlations between the features and the output.

Output layer: This layer generates the model's final output, which is often a probability distribution over dataset classes.

3.2.2 BERT for Text Classification:

A pre-trained language model called BERT (Bidirectional Encoder Representations from Transformers) was created by Google in 2018. For natural language processing (NLP) applications including text classification, question answering, and language translation, it is a sort of transformer-based neural network architecture.

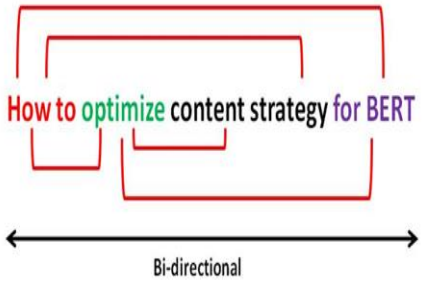
In contrast to conventional NLP models, which are trained on only one direction, BERT is dubbed bidirectional because it is trained on both the left and right context of each word in a sentence. This enables BERT to fully understand a sentence's context and generate more precise predictions.

The pre-training process of BERT involves training a large transformer-based neural network on a massive amount of text data, such as Wikipedia and other web sources. BERT can acquire a general knowledge of natural language and the relationships between words through this method, which can then be tailored for certain NLP tasks.

When using BERT for a particular NLP task, such as sentiment analysis or question answering, the pre-trained model is

refined using a smaller dataset that is tailored to the task. The pre-trained model's final layers are removed or added during fine-tuning, and the model is trained via back propagation on the task-specific dataset.

In a variety of NLP tasks, BERT has achieved state-of-the-art performance, and its pre-trained weights are publicly available, making it easy for researchers and developers to use it for their own NLP applications.



3.2.3 Embeddings:

Image Embedding:

The process of turning an image into a vector representation that can be fed into machine learning models is known as image embedding. Convolutional neural networks (CNNs) that have already been trained to extract high-level features from images, like VGG, ResNet, or MobileNet, can be used to create image embedding.

The process of image embedding involves passing an image through a pre-trained CNN, which extracts high-level features from the image. These features are then flattened and fed into a fully connected layer to generate a fixed-size vector representation of the image, which is also called an image embedding.

Text Embedding:

The process of transforming text into a numerical vector representation that may be fed into machine learning models is referred to as text embedding.Text embedding can be achieved using pre-trained language models such as BERT, GPT, or Word2Vec, which are trained to extract semantic information from text.

The process of text embedding involves passing the text through a pre-trained language model, which generates a sequence of contextualized embeddings for each word in the text. These embeddings capture the meaning and context of each word, as well as the relationships between words in the sentence.

Joint Embedding:

Joint embedding refers to the process of creating a shared vector space that can represent both images and text in a way that captures their relationships. This is achieved by combining image embedding and text embedding into a single vector space, where each image and text has a corresponding vector representation that Is close to similar images and text in terms of their meaning.

3.2.4 Similarity detection Module:

After performing all the above operations we store the joint embeddings for all the data. Now we follow the same strategy for the Input text and image which is given by the user. Now it's time to compare the similarity between the joint embedding from our dataset to the user provided text and image joint embedding.To find the similarity there are many different algorithms like Cosine similarity, Jaccard similarity, Levenshtein distance, Sequence alignment.

Here, similarity detection is accomplished using cosine similarity. A measure of similarity between two non-zero vectors in an inner product space is called cosine similarity. Between -1 and 1, it determines the cosine of the angle formed by the two vectors. When two vectors have a cosine similarity of 1, they are said to be identical, when they have a cosine similarity of 0, they are said to be orthogonal, and when they have a cosine similarity of 1, they are diametrically opposed.

In the domain of machine learning and natural language processing, cosine similarity is typically used to compare the similarity between two documents or chunks of text. In this instance, the vectors are produced by converting each document into a vector of word embeddings or a bag of words.

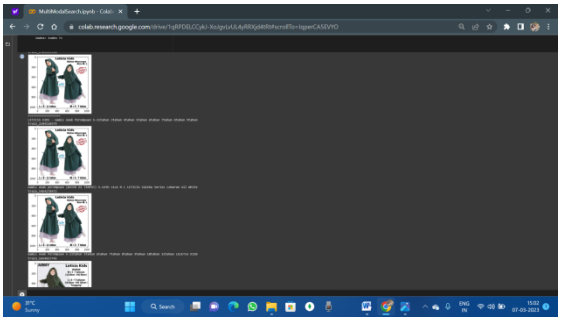
The Cosine Similarity increases as the papers' angles get closer (Cos theta).

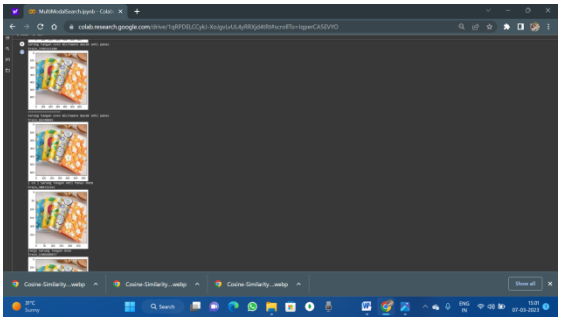
$$Cos\theta = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\,\|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2}\sqrt{\sum_1^n b_i^2}}$$

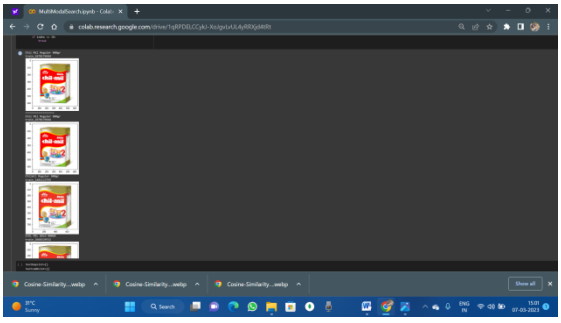
where, $\vec{a}\cdot\vec{b} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$ is the dot product of the two vectors.

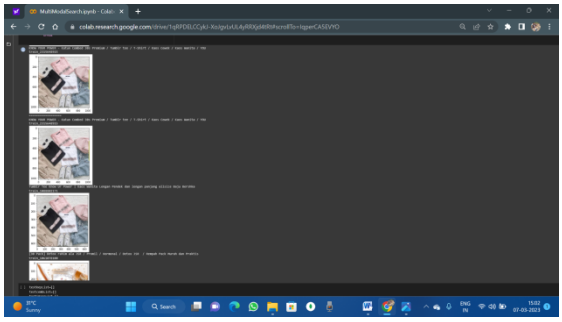
By checking the similarity between both joint embeddings from our database to user inputs we get top 'N' nearest results as output. For that we are using nearest neighbour 'kd-tree' algorithm.

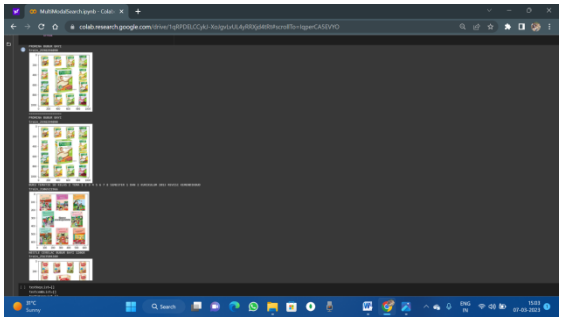
4. Result:











4. Conclusion:

Multi-model search combining text and image is a powerful technique that can enhance the accuracy and relevance of search results.By analyzing both visual and textual features, the multi-model search can provide a more comprehensive understanding of the content and context of a given query. Applications for multi-model text and picture search include visual search, image captioning, and recommendation systems. The quality of the individual models used to handle textual and visual data, as well as the efficacy of the methods used to combine them, are key factors in the success of multi-model search for text and images. The In general, multi-model search for text and images is a fascinating area of study that has the potential to greatly increase the precision and relevance of search results as well as open up new applications in a variety of fields.

5. References:

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... & Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3128-3137).

Kiela, D., & Bottou, L. (2014). Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 36-45).

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4), 600-612.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition (pp. 248-255).IEEE.

"Multi-Objective Bayesian Optimization for Neural Architecture Search" by Yingbo Zhou et al. (AAAI 2021) - This paper proposes a multi-objective Bayesian optimization approach for neural architecture search that can optimize multiple objectives simultaneously .