International Conference on Engineering & Technology (ICET-2020)

Available at: *International Journal of Scientific Research and Engineering Development*

RESEARCH ARTICLE OPEN ACCESS

Privacy Preserving Location Information Retrieval Services without Encryption

Nandhini. R[1], Geetha.T[2]

[1]*ME-Computer Science and Engineering,Dhanalakshmi Srinivasan Engineering College,Perambalur*

[2]*Assistant Professor,Dhanalakshmi Srinivasan Engineering College,Perambalur*

[1]Nandhunive2321@gmail.com

[2]Geethut14@gmail.com

----------**********----------

Abstract:

The process of recognizing desired images with information from a large collection is widely used in applications of computer vision. To improve the performance of information retrieval an efficient and accurate system is required. To overcome the problems of existing approaches, proposed an Encryption-Free framework for Privacy-preserving Image Recognition, called by EnfPire. EnfPire does not rely on cryptographic techniques so that it is not impose any restrictions on the server's recognition algorithm because it. In EnfPire, the server cannot identify the client users' current location; its candidates can only be presented. To improve the performance of image retrieval, proposed work uses various algorithms. Implement query based image retrieval with feature extraction and similarity measurement algorithms. Histogram analysis implementing to get color features from images and SIFT is implementing for retrieving shape features from given image. In proposed approach user transforms the extracted image feature ***x*** into ***y*** on the user's server and sends it to the server. With the transformation, the effectiveness of the original feature ***x*** is degraded so that the server cannot uniquely recognize the spot-ID of user from ***y***. It only retrieves the relevant spot ID's. After that, results are processed by user's server. Then unique spot ID will identify and information regarding the spot and relevant images will be shown to the user.

Index Terms—**Image Training, Image Invert and Smoothing, Feature Extraction using SIFT, Similarity Measurement, Image Recognition, Location Information Retrieval.**

----------**********----------

I. INTRODUCTION

NETWORK SECURITY BASICS

Information retrieval method has been important to the achievement of the Web. Web based indexing and searching methods along with Google and Yahoo have profoundly modified the manner of accessing records. For the semantic internet technologies to have an impact, they'll need to be compatible with Web engines like Google and information retrieval generation in popular. Here propose several approaches to the use of records retrieval systems with both semantic web files and with text files which have semantic web annotations. One prescient of the Semantic Web is that it is going to be similar to the Web that understand nowadays, besides that files might be enriched by annotations in machine understandable markup. These annotations will offer metadata about the files as well as machine interpretable statements

taking pictures some of the meaning of the files' content material.

Here explains the initial experiments that shows how existing Information Retrieval structures can be coaxed into assisting this situation using swangling technique call by to encode RDF triples as word-like terms. In an alternative approach, semantic web content material will exist in separate documents that reference and describe the content material of conventional web files. Here too it could be acceptable to apply a conventional structure which includes Google to index and retrieve those files. Here discuss how the swangling approach can also be used to add assertions to RDF files in a way this is well suited with many standard web searches.

A final approach to the use of IR engines for files is to construct custom indexing and retrieval engines in particular designed to web with semantic internet documents as adverse to standard ones. Here describe Swoogle, a prototype crawler-primarily based serps for RDF files. This permits customers to retrieve indexed RDF documents primarily based on the RDF classes and functions they use and additionally uses the Haircut information retrieval system to retrieve files using character based n-grams. The next phase will inspire the capacity to index and search for files along with or annotated with semantic web content material. Three will lay out the possible methods to adapt information retrieval systems to the Semantic Web and describe three special prototype systems that built to explore the trouble.

Physical Network Security

One frequently unnoticed element of network security includes maintaining your hardware services covered from robbery or malicious intrusion. Corporations spend big sums of money to develop their community servers, network switches and different community components in properly-guarded facilities. While those measures are not practical for homeowners, to nevertheless preserve their password-blanketed broadband routers in non-public places, away from nosy neighbours and residence visitors.

If information robbery via physical method stealing a system or router is a concern, one solution is to stop storing your facts regionally. Online backup offerings and cloud storage sites maintain sensitive files stored off-website online at a secure backup area so that even if the nearby hardware is stolen or compromised, the documents are present secure in some other place.

Physical security is much more important, because of the use of mobile gadgets. Smart phones are especially easy to leave behind or have fall out of a pocket. News stories in the press abound of people who have their smart phones stolen in public places, sometimes even while they are using them. Be alert to the physical environment on operate mobile gadgets and placed them away. If your system supports software program that permits you to track the device or remotely eliminate its information, spark off it, and use a password with the device to prevent a co-employee or acquaintance from snooping when you are out of the room.

Password Protection

If implemented well, passwords are an incredibly effective device for enhancing community security, however some humans don't take password management critically and insist on the use of weak, smooth-to-guess password like "123456" on their gadgets and networks.

Follow some common experience nice practices in password control to greatly improve the security protection on a computer network:

• Set strong passwords or skip codes on all devices that join the community.

• Change the default administrator password of network routers.

• Do now not percentage passwords with others extra regularly than essential. Set up visitor community access to for friends and visitors, if possible.

• Change passwords whilst they'll have end up recognized.

If you avoid using efficient passwords due to the fact they're more difficult to remember, store them in a password manager.

Spyware

Even without physical access to a device or knowing any network passwords, malicious programs referred to as malware can infect computers and networks. This happens when go to malicious websites by chance or through a link in a phishing email. Some monitor someone's computer utilization and web-browsing conduct to file the facts to companies who use it to create targeted advertising. Other types of malware try to access personal facts. One of the maximum dangerous sorts of malware, keylogger software program, captures and sends the history of all keyboard key presses a person makes, which captures passwords and credit card numbers. All spyware on a laptop attempts to feature without the expertise of humans using it, thereby providing a security risk. Because spyware is difficult to detect and avoid, protection experts suggest installing and maintaining official anti-adware software on laptop networks.

Online Privacy

Personal stalkers, identity thieves, and possibly even authorities agencies, reveal humans's online habits and moves well past the scope of basic malware.

Wi-Fi hotspot utilization on computer trains and vehicles screen a person's vicinity, for example. Even inside the virtual system, an lot about a person's identity may be tracked on line through the IP addresses of their networks and their social community sports.

Techniques to protect a person's privacy on line consist of anonymous net proxy servers and VPN offerings. Though retaining whole privateness on-line is not completely possible, the ones methods shield privacy to a certain degree.

II. RELATEDWORK

Mohamed Aly, et.al,.[1] proposed the idea of distributed Kd-Tree parallelization. Distributed Kd-Trees is a method for constructing image retrieval structures that may manage loads of tens of millions of pictures. It is based totally on dividing the Kd-Tree into a "root subtree" that resides on a root machine, and numerous "leaf subtrees", every living on a leaf storage. The root device handles incoming queries and farms out function matching to the correct small subset of the leaf machines. Provide the complicated details of backtracking in the DKdt in addition to a unique way to construct the tree. The implementation of IKdt with MapReduce is straightforward, see Algorithm. At training time, the Feature MapReduce is empty, while the Index MapReduce builds the independent Kd-Trees from groups of images, where the Map distributes features according to the image id, and the Reduce builds the Kdt with the features assigned to every machine. At query time, the Distribution MapReduce dispatches the features to all the M Kdts (machines). The Matching MapReduce searches the Kdts on each machine in the Map and performs the counting and sorting in the Reduce. The implementation of DKdt is outlined here. The great distinction from IKdt is the Feature MapReduce, which builds the pinnacle of the Kd-Tree. Given M machines, the highest part of the Kdt has to have dlog2Me levels, so that it has at least M leaves. The Feature Map subsamples the input features by using emitting one out of every

input bypass features, and the Feature Reduce builds the Kdt with the ones features. The Index MapReduce builds the M backside elements of the tree, where the Index Map directs the database features to the Kdt a good way to personal it, which is the primary leaf of the pinnacle part that the feature reaches with depth first seek. The Index Reduce then builds the respective leaf Kdts with the features it owns. The Matching MapReduce then plays the hunt in the leaf Kdts and the counting and sorting of images, as in IKdt.

Mohammad Norouzi, et.al,.[2] Proposed method became known as multi-index hashing, as binary codes from the database are indexed m instances into m specific hash tables, primarily based on m disjoint substrings. Given a query code, entries that fall near the query in at the least one such substring are taken into consideration neighbor candidates. Candidates are then checked for validity the use of the whole binary code, to eliminate any non-r-neighbours. To be realistic for large-scale datasets, the substrings need to be selected so that the set of candidates is small, and storage necessities are affordable. Also require that all actual neighbors will be located. The key concept right here stems from the reality that, with n binary codes of q bits, the enormous majority of the 2q possible buckets in a complete hash desk will be empty, due to the fact that 2q _ n. It seems costly to examine all L(q; r) buckets inside r bits of a question, on account that maximum of them contain no items. Instead, merge many buckets collectively (maximum of which might be empty) by marginalizing over one of a kind dimensions of the Hamming space. In the distribution of the code substring comprising the primary s bits is the outcome of marginalizing the distribution of binary codes during the last q desk includes all codes with the same three first s bits, but having any of the two remaining s bits. Unfortunately those large buckets are not confined to the Hamming extent of interest around the question. Hence not all objects inside the merged buckets are r-neighbors of the question, so we then need to cull any candidate that isn't a true r-neighbor.

Christoph Strecha, et.Al,.[3] Proposed a feature descriptors to be invariant to certain classes of photometric and geometric variations, particularly, affine and intensity scale changes. However, actual differences that an image can go through can handiest be about modeled in this manner, and for that reason most descriptors are simplest approximately invariant in exercise. Second, descriptors are typically high dimensional. In massive-scale retrieval and matching problems, this may pose challenges in storing and retrieving descriptor information. PCA has been substantially used to reduce the dimensionality of SIFT vectors. In this way, the number of bits required to explain every measurement can be reduced with out loss in matching overall performance. In, a whitening linear transform become proposed further to benefit from the performance of rapid nearest-neighbor search techniques. The three procedures above are usually unsupervised strategies and from time to time require a complex optimization scheme. Often, they are now not in particular tuned for keypoint matching and do now not typically produce descriptors as short as one could require for big-scale keypoint matching. Proposed formulation pertains to supervised metric learning strategies. The hassle of optimizing SIFT-like descriptors can be approached from the attitude of metric getting to know, wherein many efficient approaches have been currently advanced for gaining knowledge of similarity between statistics from a education set of similar and assorted pairs.

Akrivi Vlachou, et.Al,.[4] Proposed a framework for distributed metric-based similarity search that is based on a extremely good-peer structure, assuming that cooperative friends keep and index their statistics in an self sufficient manner. Each peer have to be capable of manner effectively similarity queries based on its locally stored records. Thus, each peer indexes its local records with the aid of using the M-Tree. The M-Tree includes a hierarchy

of hyper-spheres and is one of the most commonly used centralized indexing techniques for looking in metric spaces. When a peer connects to a first-rate-peer, it publishes the set of hyper-spheres saved at the foundation of its M-Tree to its fantastic-peer, as a summarization of the saved statistics. The incredible-friends save the collected hyper-spheres the use of an M-Tree index, in order to direct queries only to relevant friends efficaciously, hence setting up a peer choice mechanism. Capitalizing on their nearby metric index systems, outstanding-peers alternate precis facts to assemble metric-based routing indices, which improve the overall performance of question routing extensively. Then, given a range question, this superb peer selection mechanism permits efficient query routing only to that subset of first-rate-peers which can be responsible for friends with applicable query outcomes. The M-Tree approach builds the index in a backside-up way and the insertion approach in addition to the block length influences the satisfactory hyper-spheres of the foundation. On the other hand, iDistance is predicated on clustering and the employed clustering technique affects its usual overall performance. A standard clustering set of rules, consisting of k-means, may result in a terrible performance, at the same time as an utility-unique clustering method may also enhance the overall performance of routing. The gain of our framework is that it does not require the existence of a clustering structure within the information.

Rizwan Mian, et.Al,.[5] Proposed technique of workload execution is easy. A configuration plays a imperative position in workload execution. A search algorithm hunts for a suitable configuration given an objective which includes minimum cost. Using this configuration, the sources are made available or provisioned, and the workload is mapped on the ones sources. Finally, dynamic refinements are used to address any changes in the environment, workload or the SLOs. A customer submits a workload to the manager. The manager includes 3 additives (a) a configurator, (b) a scheduler, and (c) a provisioner. The configurator is the brains of the gadget. It determines a appropriate configuration, which includes storage and processing resources, to execute the workload against an objective. This configuration is then exceeded to the provisioning manner or the provisioner. The provisioner prepares the execution systems. It allocates the processing resources (VMs) as required by using the configuration. The provisioner also attaches statistics walls to the processing sources. In addition, the provisioner creates replicas of facts walls if needed. Once the provisioner finishes, the scheduler maps requests of the workload to the execution structures as required with the aid of the configuration, and the workload execution starts. The workload is achieved and a few comments is sent again to the manager periodically. The comments could encompass heartbeats or execution instances. The manager may also propose a brand new configuration based on the comments. Revisions to the contemporary configuration may be essential because of some of motives which include excessive SLA violations or a trade within the workload. If the deployed configuration is revised, the provisioner and the scheduler respectively modify the assets and dispatch the workload according to the brand new configuration.

III. METHODOLOGY

The purpose of PCBIR is to hide users' query image from a retrieval server as well as make the server unable to identify which gallery images are matched to the query image. This can be achieved by cryptographic techniques; that is, the users encrypt a visual feature extracted from their query image before sending it to the server and the server calculates the similarity between the visual feature of the query image and that of each gallery image in the encrypted domain. This work employed homomorphic encryption (HE), which is also used for privacy-preserving video retrieval. Focused On texture based features including lines and circles and proposed a Hough Transform-based approach to extract them from an input encrypted image. Cryptography-based PCBIR methods,

especially HE-based methods, are disadvantageous in computational efficiency. In Hash Code based approach a query image is first transformed into a hash code and a part of bits in the hash code are masked on the user side. Next, the partially masked hash code is sent to a retrieval server. The server compares the unmasked bits in the sent hash code with that of each gallery image in a database, and returns the user a set of images containing the same bits with the unmasked ones. Finally, the user screens the results returned from the server using the original hash code of the query for removing mismatched images.

IV. SECURE GROUP DATA SHARING WITH USER REVOCATION

To overcome the problems present in existing approaches here proposed an encryption-free framework for privacy-preserving image recognition called by EnfPire. In this proposed approach the client users' locations are represented as a spot-ID. Unlike this, in the mobile services based on GPS, the users' locations are represented as a numerical coordinate. There have been proposed many methodologies for protecting the location data. In the transformation-based methods, the users transform their exact location coordinate into another coordinate before sending it to the server. A typical approach is to replace the exact location coordinate with that of a certain near-by landmark such as an intersection or a building.

Here developers are provide Privacy-preserving Image search system, which is a step towards feasible cloud services which provide secure content-based large-scale image search with fine-grained access control. Users can search on others' images if they are authorized by the image owners. Majority of the computationally extensive jobs are managed by way of the cloud, and a querier can now clearly ship the query and receive the result. Specially, to address massive images, design our device appropriate for distributed and parallel computation and introduce several optimizations to further expedite the hunt procedure. Proposed system also supports privacy-preserving image storage and sharing among users. Users can search on others' images if they are authorized by the image owners. Proposed system provides the majority of the search process to the cloud, however neither the image content nor the query is discovered to the cloud. What's extra, at some stage in the search, no interaction is required among the information owner and the querier or the cloud.

Methodology

Color Feature Extraction Using Histogram

Step 1: For extracting color histogram, at first, the RGB color space is converted to HSV color space.
Step 2: The color histogram is calculated based on MPEG-7 Scalable color.
Step 3: The color space is uniformly quantized into 16 levels of hue, 4 levels of saturation and value giving a total of 256 bits.
Step 4: To lower this number and make the application scalable, the histogram is encoded using Haar transform.
Step 5: Usage of subset coefficient in Haar representation is 64 bins. Global color histogram is constructed for all images.

SIFT Feature Extraction

Processing Steps:

Step 1: Constructing a scale space: This is the initial preparation. Here create internal structure of the original image to ensure scale invariance.

Step 2: LoG Approximation: The Laplacian of Gaussian is used for finding interesting points (or key points) in an image.

Step 3: Finding keypoints: With the super fast approximation, now try to find key points. These are maxima and minima in the Difference of Gaussian image calculate in step 2.

Step 4: Get rid of bad key points: Edges and low contrast regions are bad keypoints. Eliminating these makes the algorithm efficient and robust.

Step 5: Assigning an orientation to the keypoints: An orientation is calculated for each key point.

Step 6: Generate SIFT capabilities: Finally, with scale and rotation invariance in place, one greater illustration is generated. This enables uniquely perceive capabilities. That was a top level view of the entire algorithm.

Image recognition

The new recognizer increases the server's spot-recognition performance, which is not desirable from the aspect of privacy protection. This approach should employ a transformation method that makes the server unable to judge whether visual features sent from the users are original version or transformed version. The customers have to use a simple recognizer that entails no training period due to their restrained computational resources.

Similarity Measurement Algorithm

Input: X = {x1, x2, x3,…..,xn} be the set of data points , Y= {y1,y2,y3…yn} be the set of data points and V = {v1,v2,v3,….,vn} be the set of centers

Step 1: Select 'c' cluster centers arbitrarily

Step 2: Calculate the distance between every pixels and identify cluster centers using the Euclidean Distance metric as defined below

$$Dist(X,Y) = \sqrt{\sum_{j=1}^{n}(X_{ij} - Y_{ij})^2}$$----------Eqn(6)

X, Y are the set of data points

Step 3: Where the cluster center is minimum of all cluster centers then pixel is assigned to that cluster center.

Step 4: New cluster center is calculated using

$$V_i = \frac{1}{c_i}\sum_{1}^{ci} x_i$$----------Eqn(7)

Where cluster center denoted as Vi, number of pixels in the cluster was denoted by ci.

Step 5: Then the distance between all pixel and new obtained cluster is recalculated.

Step 6: If no pixels were reassigned then stop otherwise repeat steps from 3 to 5.

Information retrieval

When receiving the visual feature from the users, the server recognizes the spot in the photo using the image recognizer and returns the corresponding information in the database to the users. Here employed similarity based recognition method on the server side, whereas employed closest detection algorithm for a recognition method on the user side.

- If I is the database image and is the query image, then the similarity measure is computed as follows,
- Calculate histogram vector vI = [vI1, vI2, ….vIn] and ccv vector called cI = [cI1, cI2, ….cIn] of the database images.
- Query is processing and Calculate the vectors vI and cI.
- The Manhattan distance between two feature vectors can then be used as the similarity measurement:
- If $d \leq \tau$ (threshold) then the images match.
- From all the matching images to display top images as a result.

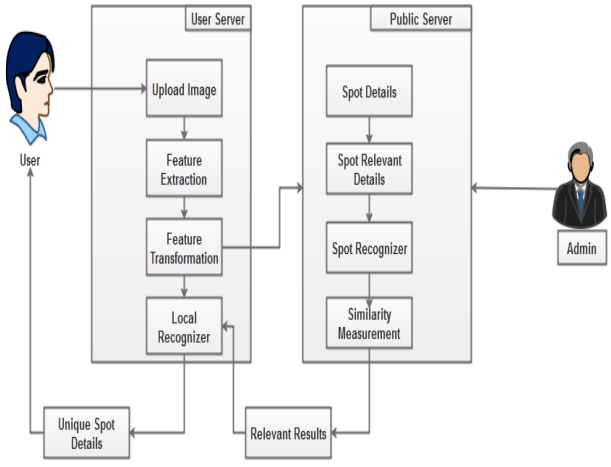

Fig 4.1: Architecture for Proposed Work

VI CONCLUSION

Proposed a EnfPire framework for photo-based information services. In such services, a server can easily recognize client users' current location. To protect the location information, EnfPire abstracts it using a feature transformation. The application performs a simple color-based search in an image database for an input query image, using color, texture and shape to give the images which are similar to the input image as the output. The number of search results may vary depending on the number of similar images in the database. The similarity metrics have been used based on distances like Manhattan distance. Different features of the image like colour, shape and text are used to extract the number of images based on the query image as input. Similarity comparison, extracting feature signatures of every image based on its pixel values and defining rules for comparing images. Distance metric or matching criteria is the main tool for retrieving similar images from large image databases for all the above categories of search. The Manhattan distance is used to determine similarities between a pair of images in the content based image retrieval application.

REFERENCES

[1] Aly, Mohamed, Mario Munich, and Pietro Perona. "Distributed kd-trees for retrieval from very large image collections." In Proceedings of the British Machine Vision Conference (BMVC), vol. 17. 2011.

[2] Norouzi, Mohammad, Ali Punjani, and David J. Fleet. "Fast exact search in hamming space with multi-index hashing." IEEE transactions on pattern analysis and machine intelligence 36, no. 6 (2014): 1107-1119.

[3] Strecha, Christoph, Alex Bronstein, Michael Bronstein, and Pascal Fua. "LDAHash: Improved matching with smaller descriptors." IEEE transactions on pattern analysis and machine intelligence 34, no. 1 (2011): 66-78.

[4] Vlachou, Akrivi, Christos Doulkeridis, and Yannis Kotidis. "Metric-based similarity search in unstructured peer-to-peer systems." In Transactions on Large-Scale Data-and Knowledge-Centered Systems V, pp. 28-48. Springer, Berlin, Heidelberg, 2012.

[5] Mian, Rizwan, and Patrick Martin. "Executing data-intensive workloads in a Cloud." In Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012), pp. 758-763. IEEE Computer Society, 2012.

[6] Jegou, Herve, Matthijs Douze, and Cordelia Schmid. "Product quantization for nearest neighbor search." IEEE transactions on pattern analysis and machine intelligence 33, no. 1 (2011): 117-128.

[7] Rafailidis, Dimitrios, and Stefanos Antaris. "Indexing media storms on flink." In Big Data (Big Data), 2015 IEEE International Conference on, pp. 2836-2838. IEEE, 2015.

[8] Hong, Yang, Qiwei Tang, Xiaofeng Gao, Bin Yao, Guihai Chen, and Shaojie Tang. "Efficient R-tree based indexing scheme for server-centric cloud storage system." IEEE Transactions on Knowledge and Data Engineering 28, no. 6 (2016): 1503-1517.

[9] Zhu, Ming-Dong, De-Rong Shen, Kou Yue, Tie-Zheng Nie, and Ge Yu. "A Framework for Supporting Tree-Like Indexes on the Chord Overlay." Journal of Computer Science and Technology 28, no. 6 (2013): 962-972.

[10] Avrithis, Yannis, Ioannis Z. Emiris, and Georgios Samaras. "High-dimensional approximate nearest neighbor: kd Generalized Randomized Forests." arXiv preprint arXiv:1603.09596 (2016).

