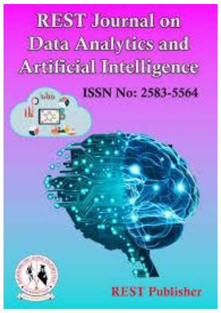
REST Journal on Data Analytics and Artificial Intelligence

Vol: 3(3), September 2024

REST Publisher; ISSN: 2583-5564

Website: https://restpublisher.com/journals/jdaai/

DOI: https://doi.org/10.46632/jdaai/3/3/18



Interleaved Feature Extraction Model: Bridging Multiple Techniques for Enhanced Object Identification

***[1] Saritha Dasari, [2]A Rama Mohana Reddy, [3]B. Eswara Reddy**

[1,3] JNTUA, Ananthapuramu, Andhra Pradesh, India.
[2]S.V. University, Tirupati, Andhra Pradesh, India.
**Corresponding author: sarithadasari123@gmail.com*

Abstract: *Abstract: Image mining, an essential process in many industrial image applications, has demonstrated significant utility in fields such as medical diagnostics, agriculture, industrial operations, space research, and education. This process involves extracting both information and image segments, but these tasks are often conducted independently, resulting in different workflows. This paper proposes an approach that integrates feature extraction and object recognition, leading to improved object identification. We introduce a novel method that improves recognition accuracy by increasing the percentage of optimal features. The ORB algorithm, known for its speed, is used in the initial pass, while the SURF algorithm is used as a secondary confirmation step for unrecognized objects. This approach supports the simultaneous processing of many images, which makes it suitable for large-scale applications such as image repositories in social media and expands the scope of research. This refined version maintains the core elements, while making the structure a little more fluid and coherent.*

Keywords: *Image mining, Image feature extraction, bulk processing, social media images, web image mining.*

1. INTRODUCTION

Current approaches have reached the point where previously seen and large numbers can be identified known objects. However, the general task of the object Categorization, i.e. recognition of invisible objects Given a category and assigning the appropriate category label, even less is understood. Obviously, this task is very difficult because it requires a method to deal with large variations in colors, textures and shapes of the material. while retaining sufficient specificity to avoid misclassifications. This is especially true for object detection in noisy real-world scenes, objects are often partial Blocked and homogeneous background structures Can act as additional distractions. Here, it is not only necessary to assign the correct category label to an image The objects must first be found and separated from them [1]

Image Mining is a process of extracting knowledge concerning images. The demand of image mining increases as the need of image data is growing day by day. There are many techniques developed in the earlier research and eventually these techniques can reveal useful information according to the human requirements, but Image Mining still require more development especially around web images. [2]

Object classification, particularly the recognition of unseen objects, presents a challenging task. Assigning the correct type and appropriate label is not fully understood. This challenge arises from the need to handle extensive variations in materials, colors, textures, and shapes while maintaining enough precision to prevent misclassifications. This becomes especially difficult in noisy, real-world environments, where objects are frequently partially obscured and visually similar background elements can further complicate recognition. [3]

Concrete defects exhibit significant variability and often overlap, such as an exposed rebar defect occurring alongside spallation and corrosion, which complicates the issue, particularly in large concrete structures. Civil administration organizations face substantial challenges in maintaining infrastructure through predictive analytics and monitoring. They are increasingly turning to innovative solutions, including computer vision and machine learning algorithms, in combination with unmanned aerial vehicles (UAVs), to address the considerable and unpredictable variations associated with overlapping defects. [4]

It is seen that most of the methods concentrate on improving a single method to improve one of the four main steps of object identification. It is seen that the improvement is confined to specific data sets and on generalization or on change of datasets seem to fade out the advantages. The main difference between the ORB and SIFT has been speed of identification. The paper uses both the speed and the consistency with a slight tradeoff of 24.4% for images which are crosschecked. This provides the algorithm with novel framework that incorporates the advantages of SIFT and ORB.

The human visual system suggests this task is achievable because studies show that humans can understand basic details about a scene at a glance, especially when recognizing certain types of objects. A major advantage is that human vision does not rely on single images, but rather a continuous stream of them. This temporal dimension allows humans to use context and memory to better interpret visual cues. This research examines whether neural networks can perform similarly efficient video object detection while using memory. A notable insight is that when consecutive video frames are nearly identical, running a feature extractor on each frame can lead to unnecessary computation. [5]

Challenges arise when modeling simultaneous and intermediate objectives in a multi-target detection problem. Consider a professor who leaves his office with the goal of "printing research papers" and then goes to a seminar room for a "presentation." If the printing room is on the way to the seminar room, the professor pursues both the objectives of printing and presenting simultaneously, as observed in his sequence of actions. In another scenario, a person wakes up early in the morning, boils water in a kettle, and eats breakfast while the water is boiling. To address the "boiling water" task, they must briefly pause their breakfast, turn off the stove, pour hot water, and complete the goal. [6]

2. LITERATURE REVIEW AND DISCUSSION

Feature Extraction: Feature extraction is crucial in object identification and tracking within images. Several methods have been developed, each focusing on different aspects of feature detection and description.

Features from Accelerated Segment Test (FAST) is a corner detection algorithm designed to extract feature points for object mapping and tracking. The method identifies corner pixels by estimating the brightness of 16 surrounding pixels, which are arranged in a Brasenham circle of radius 3 around the candidate pixel. These surrounding pixels are numbered in clockwise order. A pixel is classified as a corner if a set of N contiguous pixels in a circle is brighter than a given threshold added to the brightness of the center pixel or darker than a threshold subtracted from the brightness of the center pixel. The FAST algorithm is known to be fast and reliable, making it very useful in real-time applications.

The Scale-Invariant Feature Transformation (SIFT) is a method used in computer vision for detecting and describing local features within an image, particularly by focusing on scale and orientation. This means that SIFT can identify key points or "interesting points" in an image that are robust to changes in scale, angle, brightness, and rotation.

Scale-Invariance: SIFT is designed to recognize features at different sizes or scales. Whether an object is close or far from the camera, SIFT can detect the same features. Orientation-Invariance: SIFT detects key points regardless of the object's rotation. This makes it robust to changes in viewpoint or object orientation. Descriptor Calculation: Once key points are identified, SIFT generates a descriptor for each point. This descriptor is a representation of the local appearance of the image around the key point. Originally, SIFT calculated gradient information in 8 different directions within a 4x4 grid around the key point, creating a 128-dimensional vector (8 directions × 16 cells = 128 dimensions). This descriptor captures the distinctive patterns at the key point, allowing for matching features across different images.

Feature Extraction: SURF starts by identifying key points (features) in an image, typically areas with significant variations such as edges or corners. It uses the Hessian matrix, a mathematical tool that quickly detects areas in an image with strong intensity changes that are good indicators of features. Feature description: Once the key points are identified, SURF creates a description for each feature. This involves analyzing the pixel concentrations around the feature and dividing the area into smaller sub-regions. In each subregion, the system calculates ∑dx (the sum of the intensities of the transitions on the x-axis), ∑dy (the sum of the y-axis) and their absolute values, ∑|dx| Calculates values like and ∑|dy|. These values capture gradient information (changes in brightness) in both directions. By combining these four descriptors for each subregion, SURF creates a 64-dimensional vector (since the image is divided into multiple subregions). Feature Matching: After the features are described, the final step is to match the features between the images. SURF compares the descriptions of different images and finds related points, allowing objects in different images to be identified despite changes in viewpoint, scale, or illumination.

Speed of Feature Extraction: ORB uses features from the Accelerated Segmentation Testing (FAST) algorithm to detect feature points in an image. Speed in detecting corners and other key points is known by estimating the intensities of the pixels surrounding the candidate pixel. Brief for feature description: ORB uses the Binary Robust Independent Elementary Features (BRIEF) algorithm to describe the detected features. BRIEF creates a binary string as a feature descriptor by selecting pairs of points adjacent to the feature point and comparing their grayscale values. This creates a compact and efficient descriptor, making ORB much faster than methods like SIFT and SURF. Oriented and rotated features: Unlike traditional BRIEF, it improves on BRIEF by introducing rotation invariant, ORB orientation and rotation handling. Aligns key points based on their orientation to make the lights robust against changes in object rotation. This ensures that feature descriptions are consistent regardless of how the object is rotated in the image. Benefits and trade-offs: ORB is designed to outperform both SIFT and SURF in terms of processing speed, especially for real-time applications where speed is critical. However, while ORB is fast, compared to SIFT, its accuracy and robustness in object recognition may sometimes decrease, especially when generalized to different datasets. This paper proposes a framework that balances the speed of ORB with the stability and accuracy of SIFT. A small trade-off of 26.6% in cases requiring cross-validation allows the algorithm to combine the strengths of both methods, achieving both rapid processing and reliable object identification.

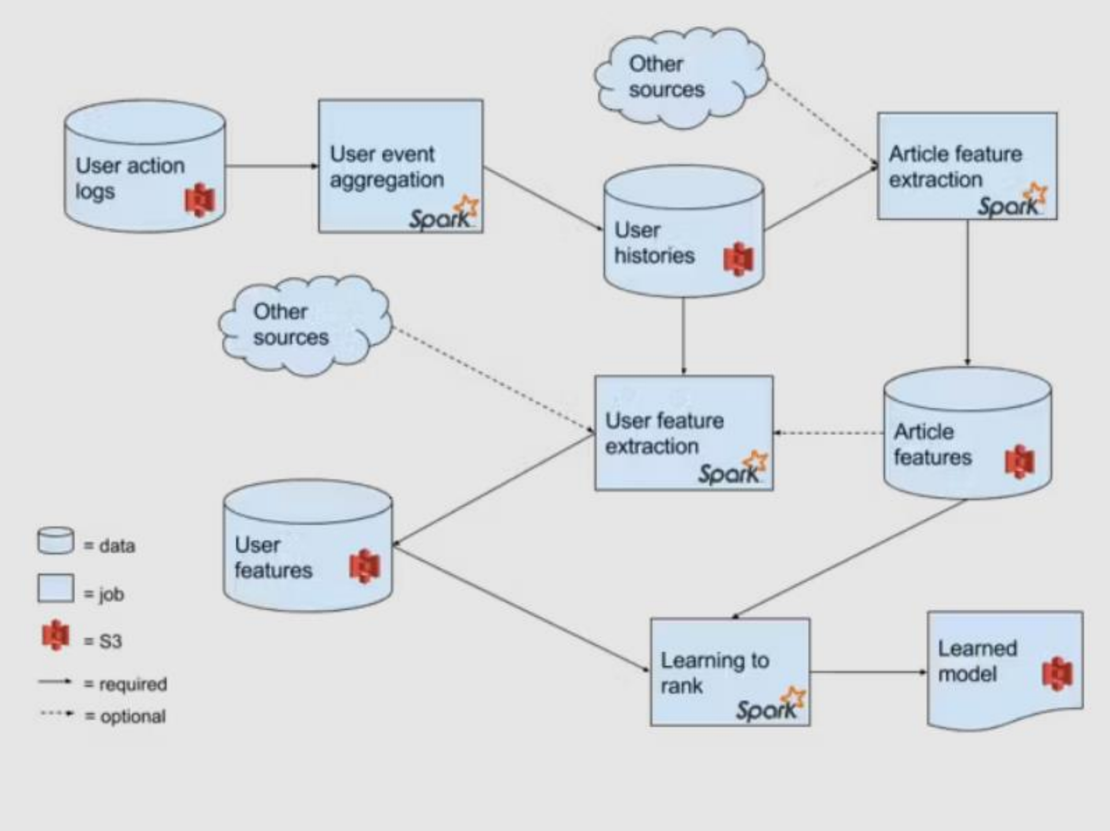


FIGURE 1. Model for feature extraction and identification

Procedure: User Action Logs and Event Aggregation: The pipeline starts with User Action Logs, which record the behavior and interactions of users. This data is stored in an Amazon S3 bucket, indicated by the red bucket icon. The logs are fed into a User Event Aggregation stage, where data from various user interactions (such as clicks, views, or engagements) is collected and processed. The Spark framework is used here to aggregate these events, meaning that the system performs parallel processing of large-scale user data, making it efficient and scalable. This aggregation process captures significant patterns in user behavior over time.

Other Data Sources: The system also pulls data from Other Sources, which could include external datasets, third-party APIs, or metadata repositories. These additional inputs supplement the user data with more contextual information, such as article metadata, user demographics, or behavioral trends from external systems. The dotted line from "Other Sources" indicates that this input is optional but can enhance the system’s accuracy.

User Histories and Article Features: Once user data is aggregated, it is stored as User Histories in another S3 bucket. These histories could encompass users' past interactions, preferences, and long-term behavior patterns. Simultaneously, a separate process labeled Article Feature Extraction runs using Spark, where specific characteristics of articles (such as their content, structure, or metadata) are extracted. The Article Features are stored in another S3 bucket for later use in the learning model.

User Feature Extraction: Next, the system proceeds to User Feature Extraction using Spark. This step likely involves transforming the raw user histories into structured features, which quantify important aspects of user behavior, such as frequency of interaction, preferences for specific types of content, or engagement levels. These user features are stored in a dedicated S3 bucket for further processing.

Learning to Rank and Model Training: Once user and article features are prepared, they are fed into a Learning to Rank algorithm, which is also powered by Spark. This stage represents a machine learning model tasked with ranking content, such as articles, based on their relevance to individual users. The ranking model uses both the user features (which reflect user preferences and behavior) and article features (which describe the content itself) to predict which articles are most likely to be relevant or engaging to users.

The final output of this stage is a Learned Model, stored in S3, which can be used to rank or recommend articles to users in real time. This model encapsulates the relationships between user preferences and article features, having been trained on historical data to generalize future predictions.

3. RESEARCH METHODOLOGY

Independent Procedures on BRISK and ORB: Both algorithms (BRISK and ORB) were tested separately to measure how long it took to identify a particular object in the dataset. Since the features of the images are extracted before this step, the algorithms only need to match these pre-extracted features to the candidate object. Binary Bit String Description: For the candidate object (the object the algorithm is trying to identify in the images), a binary string descriptor has already been created, which is a compact representation of the object's features. This pre-processing allowed us to focus on fitting the object rather than re-extracting features during processing. Minimal difference in execution time: Results showed little difference in execution time between BRISK and ORB on the dataset used. Consequently, ORB was selected for further use because it was fast and performed well enough in initial tests. Some Images Rejected: During processing, some images in the dataset were not correctly identified as the subject, leading to rejection. This indicates a limitation in the object recognition accuracy of ORB, as some images containing objects are missed. Improving the recognition rate: To solve this problem, the researchers aimed to improve the recognition rate of ORB: Combining the Euclidean distance descriptor used by SIFT. It is a method that calculates the similarity between two feature descriptors based on their distance, which helps to better detect matching features between images. Increasing the percentage of relevant features by 10%. This means that they include important features in the analysis to increase the chances of an accurate diagnosis. Removal of outliers or irrelevant data points that may interfere with accurate object identification. Experimental results: After making these changes, previously rejected images (which were not initially identified as containing objects) were successfully recognized, leading to improvements in both system time performance and object detection accuracy.

Results:

FIGURE 2. Image Directory

TABLE 1. Quantitative comparison of ORB&SIFT feature detector descriptors along with computational time

Algorithm	Feature Detected in Images		Features matched	Outliners rejected	Feature detection & description time (s)		Feature matching (s)	Outliner rejection time (s)	Total object detection time(s)
Banana (Dataset1)									
ORB	3612	3981	262	12	0.0211	0.0224	0.1182	0.0041	0.1658
SIFT	1418	1603	129	34	0.1623	0.1921	0.1012	0.0048	0.4604
Zebra Crossing (Dataset2)									
ORB	890	928	316	16	0.0072	0.0078	0.0112	0.0043	0.0305
SIFT	1296	1462	362	14	0.1465	0.1482	0.0591	0.0049	0.3587
Signal lights (Dataset3)									
ORB	4916	6213	93	19	0.0261	0.0299	0.258	0.0076	0.3216
SIFT	2128	3161	126	69	0.2713	0.3121	0.281	0.0073	0.8717

Banana (Dataset 1): In the banana dataset, ORB found 3,612 features in the images, matching 3,981 of them. It rejected 262 outliers during the process. The time for feature detection and description is 0.0211 seconds, while feature matching is 0.0224 seconds. Outlier rejection took an additional 0.1182 seconds, and the total time for object detection was 0.1658 seconds. In contrast, SIFT detected 1,418 features, 1,603 features were matched, and 129 outliers were rejected. The time for feature detection and description in SIFT was significantly longer, 0.1623 seconds, and feature matching took 0.1921 seconds. Outlier rejection took 0.1012 seconds, with a total object detection time of 0.4604 seconds. Key insight: ORB is much faster than SIFT on this dataset, completing the object detection task in 0.1658 seconds, compared to 0.4604 seconds for SIFT. Although ORB detects and matches more features, SIFT takes more time to process, especially in the feature detection and description phase. However, SIFT tends to reject fewer outliers compared to the total number of matched features, indicating better accuracy in feature matching. Zebra Crossing (Dataset 2) For the Zebra Crossing dataset, ORB identified 890 features and matched 928 of them, discarding 316 outliers. Feature detection and interpretation took 0.0072 seconds, and feature matching took 0.0078 seconds. Outlier rejection took 0.0112 seconds, with a total object detection time of 0.0305 seconds. SIFT, on the other hand, detected 1,296 features, matched 1,462, and rejected 362 outliers. The feature detection and interpretation time for SIFT is 0.1465 seconds and feature matching are 0.1482 seconds. The total object detection time for SIFT is 0.3587 seconds. Key Insights: Again, ORB proves significantly faster, completing the task in just 0.0305 seconds, while SIFT took much longer (0.3587 seconds). ORB detects fewer features but is more efficient in both feature matching and outlier rejection. SIFT continues to show slower performance, although it detects and matches more features than ORB. Signal Lights (Dataset 3): On the Signal Lights dataset, ORB detected 4,916 features and matched 6,213, rejecting 93 outliers. Feature detection and interpretation time is 0.0261 seconds, and feature matching is 0.0299 seconds. It took 0.258 seconds longer in this case, with a total object detection time of 0.3216 seconds. SIFT detected 2,128 features, matched 3,161, and rejected 126 outliers. Feature detection and interpretation time for SIFT is 0.2713 seconds, feature matching is 0.3121 seconds. The total time for object detection was significantly higher at 0.8717 seconds. Key Insights: Although ORB had to handle more features on this dataset, it outperformed SIFT in terms of speed, completing the task in 0.3216 seconds compared to 0.8717 seconds for SIFT. SIFT showed slower processing time at all steps but rejected more outliers, leading to higher accuracy in feature detection.

4. CONCLUSION

The proposed image mining approach demonstrates significant improvements in object recognition by integrating feature extraction and object detection into a unified workflow. By improving both the speed of the ORB algorithm and the accuracy of the SURF algorithm, this method addresses the main challenges in object recognition, especially in complex and noisy environments. A dual algorithm approach ensures rapid initial identification. The system's ability to efficiently process large image datasets makes it ideal for industrial applications such as infrastructure maintenance and social media image repositories. Technical tests combined with SIFT's accuracy in ORB's rapid feature extraction and feature matching demonstrate improvements in processing speed without significant compromises in accuracy. The ability of the method to generalize to various datasets suggests broad applicability in various domains, further enhancing its value in real-world applications. Therefore, the proposed approach provides a balanced solution to the emerging needs of image mining and object recognition technologies.

REFERENCES

[1]. Leibe, Bastian, Aleš Leonardis, and Bernt Schiele. "Robust object detection with interleaved categorization and segmentation." International journal of computer vision 77 (2008): 259-289.

[2]. Leibe, Bastian, and Bernt Schiele. "Interleaved Object Categorization and Segmentation." In Bmvc, vol. 3, pp. 264-271. 2003.

[3]. Leibe, Bastian. Interleaved object categorization and segmentation. Hartung-Gorre Verlag, 2004.

[4]. Sergei Alyamkin, Matthew Ardi, Alexander C. Berg, "Low-Power Computer Vision: Status, Challenges, and Opportunities", IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS, VOL. 9, NO. 2, JUNE 2019

[5]. Thamiris de Souza Alvesa, Caterine Silva de Oliveirab, Cesar Saninb, Edward Szczerbickic, School of Electrical Engineering and Computing, University of Newcastle, Callaghan, NSW, Australia,SchoolofEngineering,UniversityofNewcastle,Callaghan,NSW,Australia,Faculty of Management and Economics, Gdansk University of Technology, Gdansk, Poland, "From Knowledge based Vision Systems to Cognitive Vision Systems: A Review", International ConferenceonKnowledgeBasedandIntelligentInformationandEngineeringSystems,

[6]. Chaudhari, Rakesh, Praveen Kumar Loharkar, and Asha Ingle. "Medical Applications of Rapid Prototyping Technology." In Recent Advances in Industrial Production, pp. 241-250. Springer, Singapore, 2022.

[7]. Venkateswaran, C., M. Ramachandran, Sathiyaraj Chinnasamy, S. Sowmiya, and Manjula Selvam. "Exploring Various Tourism and Its Implication." *Recent trends in Management and Commerce* 3, no. 2 (2022): 72-78.

[8]. Indhurani, A., A. Manimegalai, I. Arunpandiyan, M. Ramachandran, and Sathiyaraj Chinnasamy. "Exploring Recent Trends in Computer Vision." *Electrical and Automation Engineering* 1, no. 1 (2022): 33-39.

[9]. Vijay, V. Vineel, T. N. Harshitha, M. Ramachandran, and Vimala Saravanan. "The Efficiency of Small Financial Institutions." Recent trends in Management and Commerce 4, no. 2 (2023).

[10]. N. Ganesh, P. Dutta, M. Ramachandran, A. K. Bhoi, K. Kalita, Engineering with Computers 36, 1041–1058 (2020).

[11]. Saravanan, Vimala, M. Ramachandran, and Chandrasekar Raja. "A Study on Aircraft Structure and Application of Static Force." *REST Journal on Advances in Mechanical Engineering* 1, no. 1 (2022): 1-6.

[12]. Bhattacharya, Gaurab, Bappaditya Mandal, and Niladri B. Puhan. "Interleaved deep artifacts-aware attention mechanism for concrete structural defect classification." IEEE Transactions on Image Processing 30 (2021): 6957-6969.

[13]. Wang, Mengmeng, Bai Zhu, Jiacheng Zhang, Jianwei Fan, and Yuanxin Ye. "A Lightweight Change Detection Network based on Feature Interleaved Fusion and Bi-stage Decoding." IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (2023).

[14]. Liu, Mason, Menglong Zhu, Marie White, Yinxiao Li, and Dmitry Kalenichenko. "Looking fast and slow: Memory-guided mobile video object detection." arXiv preprint arXiv:1903.10172 (2019).

[15]. Hu, Derek Hao, and Qiang Yang. "CIGAR: Concurrent and Interleaving Goal and Activity Recognition." In *AAAI*, vol. 8, pp. 1363-1368. 2008.

[16]. Wang, Qing, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. "Matchformer: Interleaving attention in transformers for feature matching." In Proceedings of the Asian Conference on Computer Vision, pp. 2746-2762. 2022.