A NOVEL APPROACH FOR IMPROVING THE QUALITY OF DATA USING AGGREGATION MECHANISM

by

Shadi Ahmed Al-khateeb

Bachelor of Science, Yarmouk University, 2003

Master of Science, Youngstown State University, 2014

Submitted to the Graduate Faculty of

School of Computing and Information in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2021

UNIVERSITY OF PITTSBURGH

SCHOOL OF COMPUTING AND INFORMATION

This dissertation is presented

by

Shadi Ahmed Al-khateeb

It was defended on

February, 2021

and approved by

Vladimir Zadorozhny, PhD., Professor, Information Sciences

Paul Munro, PhD., Associate Professor, Information Sciences

Konstantinos Pelechrinis, PhD., Associate Professor, Information Sciences

John Grant, PhD., Professor, Computer Science, University of Maryland

A NOVEL APPROACH FOR IMPROVING THE QUALITY OF DATA USING AGGREGATION MECHANISM

Shadi Al-khateeb, PhD

University of Pittsburgh, 2021

Due to the inception of the big data applications, it is becoming increasingly important to manage and analyze large volumes of data. However, it is not always possible to efficiently analyze very big chunks of detailed data. Thus, data aggregation techniques emerged as an efficient solution for reducing the data size and providing summary of the key information in the original data. For example, yearly stock sales are used instead of daily sales to provide a general summary of the sales. Data aggregation aims to group raw data elements in order to facilitate the assessment of higher-level concepts. However, data aggregation can result in the loss of some important details in the original data, which means that the aggregation should be done in a creative manner in order to keep the data informative even if there is a loss in some details. In some cases, we may have only aggregated versions of the data due to the data collection constraints as well as high storage and processing requirements of the big data. In these cases, we need to find the relationship between aggregated datasets and original datasets. Data disaggregation is one solution for this issue. However, accurate disaggregation is not always possible and easy to utilize.

In this dissertation, we introduce a novel approach to improve the quality of data to be more informative without disaggregating the data. We propose information preserving signature based preprocessing strategy, as well as an aggregation-based information retrieval architecture using signatures. We compensate the loss of details in the raw data by highlighting the most informative parts in the aggregated data. Our approach can be used to assess similarity and correspondence between datasets and to link aggregated historical data with most related datasets. We extended our approach to be used with time series datasets. We also created hybrid signatures to be used at any aggregation level.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

First, I would like to express sincere appreciation to my advisor, Dr. Vladimir Zadorozhny, for his help, advice, and persistent encouragement throughout my Ph.D. study. Without his support and guidance, none of the work presented in this dissertation proposal would have been possible. Moreover, I am deeply indebted to him for showing me how to do successful research and how to communicate research results effectively.

I would like to thank my committee members. Their advice and comments were very constructive and helpful. I really appreciate their help and support.

Finally, I would like to express my deep gratitude to my family, and especially my parents and wife, for their love and support. I am grateful beyond words for all that they have given me.

1.0 INTRODUCTION

Due to the inception of the big data applications, it is becoming increasingly important to manage and analyze large volumes of data. However, it is not always possible to efficiently analyze very big chunks of detailed data. Thus, we need to analyze a less detailed version of the data, which should be reasonably informative. So, data aggregation techniques emerged as an efficient solution for reducing the data size and providing a summary of the key information in the original data [1]. Data aggregation aims to group raw data elements in order to facilitate the assessment of higher-level concepts. However, data aggregation can result in the loss of some important details in the original data, which means that the aggregation should be done in a creative manner in order to keep the data informative even if there is a loss in some details. Therefore, it can efficiently affect several applications that require data processing, such as indexing. In some cases, we may only have aggregated versions of the data due to the data collection constraints. In these cases, and in order to process these aggregated data we need to disaggregate the data. However, accurate disaggregation is not always possible and easy to utilize.

In our approach, we propose to improve the quality of aggregated datasets by combining the raw data and aggregation sustainable data signatures. Our approach is generic and can be applied to many domains.

More specifically, in this research we develop a scalable approach that aims to improve the quality of aggregated data by creating representative data signatures, which utilizes specific patterns around data cells.

This thesis aims to answer the following accompanied research questions:

Research Question 1: How much information can be preserved in aggregated data and how this information can be utilized?

Research Question 2: How to relate the information at different aggregation levels and how to build an efficient retrieval architecture on top of aggregated datasets?

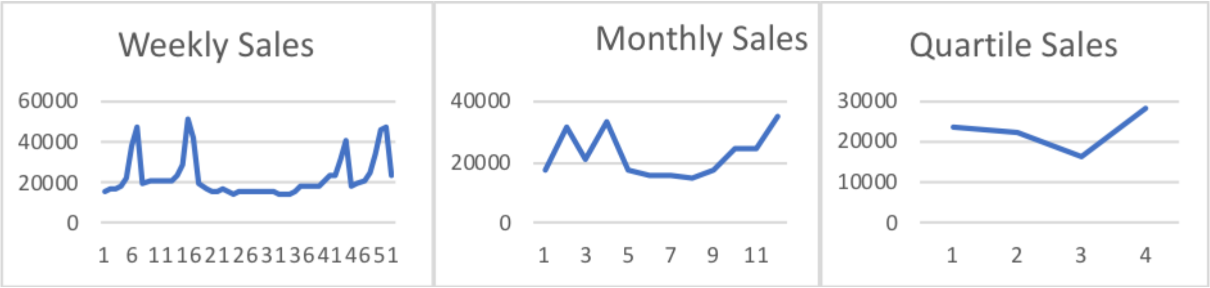
Research Question 3: How to build an efficient retrieval architecture on the top of aggregated time series datasets?

2.0 BACKGROUND AND LITERATURE REVIEW

2.1 DATA AGGREGATION

Data aggregation techniques emerged as an efficient solution for reducing the size of data and providing summary of the key information in the source data. Data aggregation aims to group raw data elements in order to facilitate the assessment of higher-level concepts. There are several methods for data aggregation [2] , such as simple arithmetic methods [3], which include averaging, central-cell, median, nearest neighbor, bilinear, bicubic. All these methods extract a value from a n x n window in the original data image as the pixel value in the new image. Another way for data aggregation is geo-statistical method [4], which considers the spatial properties in the operation of aggregation, including variance-weighted, geo-statistical variance estimation, spatial variability-weighted and simulation methods. The transform-based aggregation [5] decomposes the original dataset into components with different frequencies, in which the low-frequency components together compose a smoothed dataset. Data aggregation can result in the loss of some important details in the original data.

Figure 1 shows Walmart sales dynamics for an anonymous item. As shown in Figure 1(a), we can find the most active week(s) during the year. In Figure 1(b), the data are aggregated by month. In this figure, we can find that the best months are Feb, April, and December. However, we cannot determine the best week in each month. In Figure 1(c) the data are aggregated by quartile. In this figure, we can find that the best quarter is the fourth one that includes October, November, and December. However, we cannot determine the best month(s) among these months or the best week during the best month. From this example, we can see that aggregation can help in finding some features and patterns which are hard to find in the individual data values. On the other hand, aggregation reduces the data size. For example, if we are representing the data as a table, then the weekly data will be of size 52 X 1, the monthly data will be of size 12 X 1, and the quartile data will be of size 4 X 1.



(a) **(b)** **(c)**

Figure 1. Sales for Anonymous Item for Anonymous Walmart Store in 2011

https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/data

One of the advantages of data aggregation is improving the response time of the queries. Therefore, using aggregate data can improve the queries to be executed in a shorter time compared to the whole dataset. For example, as shown in Figure 1, to get any sales about a certain week, we need to access a dataset of size 4 (using a quartile dataset) while by using the whole dataset, we need to access a dataset of size 52. The efficiency improvement can be more notable in the large-scale datasets, where the aggregation will result in lower resource consumption including memory and CPU. Additionally, it will save the time by minimizing the search indices.

The loss of information caused by the aggregation can be attenuated using different techniques such as max-pooling, low pass filtering and wavelet decomposition.

Deep learning methodology aims to extract high level features from complex datasets. Convolutional neural networks (CNNs) are special type of artificial neural networks, in which they learn meaningful features in an adaptive manner [6]. CNN includes both features extraction and classification processes that require multiple convolutional and pooling layers to get the hierarchical properties of the input data. Max pooling is one of the widely used pooling methods. It aims to down sample the input data and reduce its dimensionality. [7] However, max pooling is quite simple and doesn't always provide optimal solution [8].

Low pass filter is a linear algorithm that is widely used as a preprocessing step in the applications of signal processing. It aims to remove the high frequency components of noise, which don't interfere with the signal spectrum [9].

Wavelet is a multi-stage process that can be used to detect sudden transitions. It captures frequency and location information at the same time, which means that it can provide us with more details about the dataset that in turn helps to create a representative signature of the dataset [10]

2.2 DATA DISAGGREGATION

Disaggregating data is one significant approach to reveal patterns that can be masked through larger aggregated data. It can help to ensure that resources are spent on the areas where they are most needed and can have the biggest impact [11]. Steady-state edge detection, harmonic analysis [12], and transient state [13] are examples of data disaggregating algorithms.

Although disaggregated data can be more informative, e.g., to train neural networks, it is always challenging to use data disaggregation techniques. For example, it can be hard to detect patterns from small disaggregated data, and if we have different data sources, these sources may have different definitions or break down of the data, which can result in biased results. There are several methods for disaggregating data, such as the method presented in [14], which performs information reconstruction from consecutive and non-overlapping summaries (histograms) by maximizing an entropy measure. However, it is not clear how this method can handle overlap or missing values.

Given low frequency timeseries such as annual sales, weekly stock and market index, the goal of temporal disaggregation is to produce a high-resolution series [15-17] such as, quarterly sales, daily stock market index, while satisfying temporal aggregation constraints, which aim to ensure that the sum, average, and the first or the last value of the resulting high frequency time series is consistent with the low frequency series. If they are consistent, related series observed at the required high frequency can be used to disaggregate the original observations. These series are called indicators. However, we should take into consideration the selecting indicators since two strongly correlated low frequency time series may not be correlated at a higher frequency [18]. Therefore, choosing good indicator series is not a straightforward task. Temporal disaggregation methods have been used for the cases of non-overlapping aggregated reports and cannot be directly applied.

One method to find an approximate solution of an under-determined linear system corresponding to the task of disaggregation is to apply least squares method (LSQ) and Tikhonov regularization [19, 20], by introducing additional constraints such as smoothness in spatial or temporal domain to allow the reconstruction to represent some parts of the target data. Although

Tikhonov regularization has been widely used in solving problems in various communities, the application of Tikhonov regularization has not been addressed in historical data fusion domain [21].

Another method to recover information from summary data is to use methods of the inverse problem theory in order to inject a priori knowledge about the domain, and finally transforming the problem into a constrained optimization problem [14]. The method shows that for smooth enough distributions, it is possible to have full recovery of information given partial sums. Although, this method could handle overlaps and missing values, the method is unable to efficiently handle data conflicts.

H-FUSE is another method that efficiently reconstructs historical counts from possibly overlapping aggregated reports [21]. It recovers times sequence from its partial sums by formulating it as an optimization problem with various constraints. The method allows the injection of domain knowledge such as smoothness and periodicity.

ARES (Automatic REStoration) is an efficient approach that automatically reconstructs and recover historical data from aggregated reports in two phases [22]: (1) estimating the sequence of historical counts using domain knowledge; (2) using the estimated sequence to derive significant patterns in the target sequence in order to refine the reconstructed time series.

Given all the previously mentioned methods for data disaggregation, we can find that each method has its own limitations. Additionally, the original data can’t be correctly retrieved, which makes it hard for the machine learning algorithms to achieve high matching accuracy. In our approach, we focus on making the aggregated data to be more informative and have details and signature without the need to use any disaggregation method.

2.3 IMAGE RETRIEVAL

We will explore our techniques in the context of two-dimensional data sets, which can be considered as images. Currently, there is a tremendous increase in the number of digital images that have been uploaded into different archive and online database. Most of the traditional method to retrieve relevant images rely on the text-based approaches, which are complex and time intensive since they rely on certain captions and metadata. Therefore, it is becoming increasingly important to find an efficient technique to retrieve relevant images from certain archives or database. Content-based image retrieval (CBIR) and image classifications are emerging approaches that aim to bridge the gap between the image feature representation and human visual understanding.

Image classification is the process of finding the most accurate specifications of the image that can be further used to classify other images into a definite number of classes [23]. Image retrieval methods can be classified into a number of categories [24] including, text based image retrieval, content based image retrieval, sketch based image retrieval, query based image retrieval, semantic based image retrieval, annotation based image retrieval.

Text based image retrieval methods depend on adding metadata to the images, such as caption, descriptions or keyword, which help in retrieving the image through the use of annotation words. However, these methods are very complex and time and resource consuming since they require a number of employees to do the manual annotation [25]. In the semantic based image retrieval, the semantic gap can be defined as the lack of synchronization between the extracted information from the visual data and the interpretation of the same data [26]. Sketch based image retrieval algorithms use sketches as an input to the algorithm, in which the sketches can be used to retrieve all related images [24].

Content Based Image Retrieval (CBIR) is considered as one of the major strategies for retrieving and classifying images. CBIR is heavily dependent on the domain of the image [27], which can be either narrow domain such as retina, fingerprint or face recognition, or broad domain such as internet images. Color, shape and textures are very important features that help to define high level semantic in the image retrieval process. Therefore, CBIR depends on analyzing these image’s features including, color, size, shape and texture, which provide better image indexing and higher accuracy in retrieving the images [28]. There are several color features that can be used to retrieving images including [29], co-occurrence matrix, difference between pixels of scan pattern and color histogram for k-mean, color covariance matrix, color histogram, color moments, and color coherence vector. Texture features represent the shape distribution. Additionally, texture representation methods can be categorized into three categories [30] including, structural, multi resolution filtering, and statistical methods. To identify a certain texture in an image, the image needs to be modeled as a two-dimensional gray level variation.

Several content-based image retrieval (CBIR) algorithms have been developed. Krishna et al study [31] provided an image indexing algorithm that utilizes k-mean algorithm. This algorithm starts with reading the image and then separating the colors using decorrelation stretching. The next step is the conversion of the RGB to L*a*b color space and finally the classification of color space under a*b* through the use of the k-means algorithm in order to separate objects. Syam et al study [32] provided a genetic algorithm that aims to extract image features and thus measure image similarity. The Gabor wavelet transform and HSV color histogram in CBIR is an approach that uses both texture feature and color histogram for quick and efficient retrieval of relevant images from the image database [33]. In this approach, researchers compute the mean and standard deviation on each color band of the image and sub-band of different wavelets. In the next step, the standard Wavelet and Gabor wavelet transforms are used to decompose the image into sub-bands.

Authors in [34] developed a new algorithm to retrieve low quality images from generic databases. Their method comprises several steps including cluster section, threshold value computing, binary images transformation, feature vector extraction, final feature vector, comparison of feature vector, and image retrieval. Measuring distances between images is another strategy of images classification [35]. Authors in [23] developed an algorithm that is able to measure distances between images by transforming each image into sequence of characters and then calculate the LZ-complexity and the string distance measure.

There are several objective image quality metrics that aim to provide some quantitative measures to estimate the quality of the image [36]. The mean squared error (MSE) is the simplest metric, which can be calculated by averaging the squared intensity differences of distorted and reference image pixels, with the related quantity of peak signal-to-noise ratio (PSNR). The Structural SIMilarity (SSIM) index is an accepted standard for image quality metrics [37]. SSIM is a method that aims to assess the similarity between two images. It also aims to predict the quality of the digital image [36]. SSIM takes into consideration the image degradation as an important change in the structural information. It also takes into consideration other factors including and contrast masking terms and luminance masking.

In some cases, users need to retrieve images form very large databases or repositories, which is considered as a complex process. Therefore, deep learning algorithms can be used to expediate the process of image retrieval. The term frequency-inverse document frequency (TF-IDF) was introduced for content based image retrieval [38].

2.4 PERFORMANCE MEASURES

The performance of classification algorithms can be measured using a number of measures including, accuracy, precision, recall and F1 score. In order to define each one of these measures, we need to introduce the confusion matrix, true positives, true negatives, false positives, false negatives. The confusion matrix can be defined as a table of rows and columns, in which each column represents the predicted class and each row represents the actual class. It aims to visualize the performance of the classification process [39]. Table 1 shows a confusion matrix that includes three classes: cat, dog and horse. In this example, the classification algorithm can correctly predict 10 cat images out of 80. On the other hand, it can wrongly predict 50 cat images as dog and 20 cat images as horse. True positive (TP) represents the number of correctly identified objects [40]. For example, true positives of cat object is 10. True negative (TN) represents the number of correctly predicted negative values [40]. For example, true negatives of cat object is 94= (9+15+30+40). On the other hand, false positive (FP) represents the incorrect positive classification [40]. For example, false positives of cat object is 25=(5+20). False negative (FN) represents the number of incorrect negative classification [40].

Table 1. Confusion Matrix

Actual Class	Predicted Class			
		Cat	Dog	Horse
	Cat	10	50	20
	Dog	5	9	15
	Horse	20	30	40

As we mentioned earlier, the performance of classification algorithms can be measured using a number of measures including, accuracy, precision, recall and F1 score [40]. Accuracy is the ratio of correctly classified objects to the total number of objects that need to be classified, which equals (TP+TN) / (TP+FP+FN+TN). Precision is the ratio of correctly predicted positive classifications to the total predicted positive classifications, which equals (TP) / (TP + FP). Recall is the ratio of correct positive classifications to the total number of positives, which equals (TP) / (TP+FN). F1- score is calculated using precision and recall, which equals

2 (Recall x Precision) / (Recall + Precision) = 2TP / (2TP + FP+ FN)

3.0 PROPOSED APPROACH

In this chapter, we describe the main challenges that we propose to tackle in this thesis as well as the proposed solutions, the assumptions, and the thesis contribution.

3.1 OVERVIEW

As discussed earlier, data can be either in a raw form or aggregated at different levels. Although, aggregation allows to speed up processing of big data, it may lead to the problem of missing some major details, which in turn can affect the quality of data. For example, in the case of indexing, the accuracy of indexing can be affected. On the other hand, raw data include o lot of details and not all these details are important. Thus, a major objective of this proposal is to improve the quality of data making it more informative by highlighting the most important parts of the data regardless of whether the data is raw or aggregated.

Data aggregation can result in the loss of some important details in the original raw data, which in turn can affect the process of indexing these data. In our approach, we could compensate the loss of details in the raw data by highlighting the most important features in the aggregated data, which helps the indexing process to get higher accuracy. One solution of the problem of the loss of some important details is to disaggregate the data. However, as we mentioned it in the previous chapter, this is not always efficient and easy to utilize.

As mentioned earlier, there are different methods for disaggregation. Each method has its own limitations. In our approach, we do not require to disaggregate. Instead, we create a signature for the aggregated data to be used instead of the aggregated data.

METHODOLOGY AND PROPOSED SOLUTIONS

Next, we address the main challenges that are related to every research question, we also describe the finished and unfinished tasks for every research question.

3.2 RESEARCH QUESTION 1: HOW MUCH INFORMATION CAN BE PRESERVED IN AGGREGATED DATA AND HOW THIS INFORMATION CAN BE UTILIZED?

3.2.1 Information preservation in aggregated data

Although data aggregation is useful for data analysis, data aggregation can lead to the loss of some important details. Given a dataset in an aggregated format that may involve some missing details based on the degree of the aggregation, our task is to make the aggregated version of the data as informative as possible without the need to disaggregate the data. In order to solve this challenge, we could detect the changes in the aggregated data and then assess the degree of these changes and consider the most significant changes as a signature of the data. The signature design addresses whether this signature will be representative enough for the aggregated data or not.

We developed an approach to highlight the most informative parts of the dataset. Additionally, we could estimate how much data quality we can preserve in the aggregated data. As a result, we could identify a data object and the relationship between different data objects. For example, if we have different aggregated versions of different images, our approach can identify each aggregated image and to whom it belongs. The identification is done by comparing the signatures of the images. As shown below in Figure 2, we have three different images with aggregated versions for each one. It is apparently hard to match the aggregated versions and the original ones especially when the aggregation is done at high level. Additionally, bitwise comparison is very hard since the original and the aggregated versions are completely different from each other, and there is no way to find some clear patterns or features.

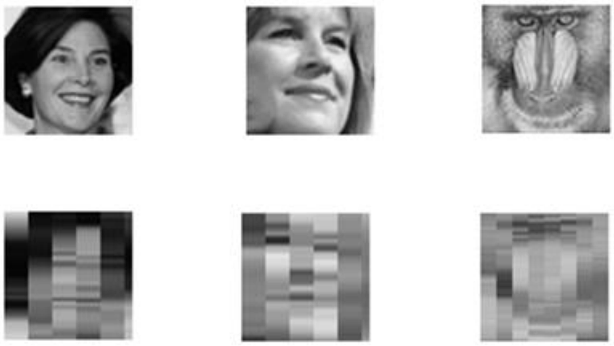


Figure 2. Images with their aggregated form

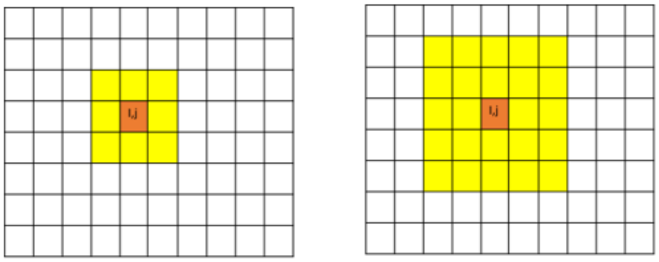
We addressed this question by creating a signature that assesses the changes around a data cell. More changes around the cell reflect higher importance of the cell. When the aggregation includes the whole side (left, right, upper, lower), this will be similar to edge detection in the field of image processing. However, in our work, we divide each side to sub-areas, then we aggregate each sub-area instead of the whole side as in the edge detection.

We expect that the signature can be used instead of original aggregated datasets. The indexing process using signature space could achieve higher accuracy in retrieving images using original space. Meanwhile, using the same method to identify corresponding images from aggregated data could not achieve the same level of accuracy.

In order to create our signature of a dataset m x n, we proposed to use a filter that is m'x n' matrix and its coefficients total that equals to 0. Therefore, the upper bound sign will have an opposite sign of the lower bound. Additionally, the left and right bounds will have opposite signs. The coefficients of the filter get lower value as the data cell being more far away from the central cell in order to give closer cells more weight than remote cells. After that, we calculated the net value of horizontal and vertical components using equation 3. We call the resulted dataset conflict matrix. Then we normalized the results using the maximum value. So, the output is in the range of [0 1]. If we want to be more localized, the matrix will be divided into segments and the normalization will be done using the maximum value of each segment. We then use a threshold value that is a value between 0 and 1. For example, when we use a threshold value of 0.6, then everything in the normalized results that is below 0.6 will be changed to 0. Higher threshold values mean that we are interested in the most informative parts of the dataset.

Figure 3 shows an example of 3 x 3 filter and 5 x 5 filter. The coefficients of the 5 x 5 filter in the x and y directions are shown below the matrix. If the filter is m' x n' and the dataset is m x n, then the following condition should be satisfied:

$$\mathbf{3 \leq m' \leq m \ \ and \ \ 3 \leq n' \leq n}$$



-1/2.83	-1/2.23	0	+1/2.23	+1/2.83
-1/2.23	-1/1.41	0	+1/1.41	+1/2.23
-1/2	-1	0	+1	+1/2
-1/2.23	-1/1.41	0	+/1.41	+1/2.23
-1/2.83	-1/2.23	0	+1/2.23	+1/2.83

+1/2.83	+1/2.23	+1/2	+1/2.83	+1/2.23
+1/2.23	-+/1.41	+1	+1/2.23	-1/1.41
0	0	0	0	0
-1/2.23	-1/1.41	-1	-1/2.23	-1/1.41
-1/2.83	-1/2.23	-1/2	-1/2.23	-1/2.83

Figure 3. Filter Coefficients for X and Y respectively

To get a certain cell value in the signature, the convolution filter will be applied on that cell.

$$\text{Cell Conflict\_X}=f(x) = \sum_{All\ k,h}\left(Filter\_X_{k,h} * \ Cell_{k,h}\right) \qquad \text{Equation 1}$$

$$\text{Cell Conflict\_Y}=f(x) = \sum_{All\ k,h}\left(Filter\_Y_{k,h} * \ Cell_{k,h}\right) \qquad \text{Equation 2}$$

$$\text{Cell Conflict} = \sqrt{Cell\ Conflict\_X^2 + Cell\ Conflict\_Y^2} \qquad \text{Equation 3}$$

Where k, and h are the dimensions of the filter in both directions.

To illustrate our approach, consider the Mandrill image from Figure 4. After applying a filter of size 31 x 31 and using a threshold value of 0.5 on each pixel, the resulted matrix will be the signature of that dataset as shown in Figure 4.b.

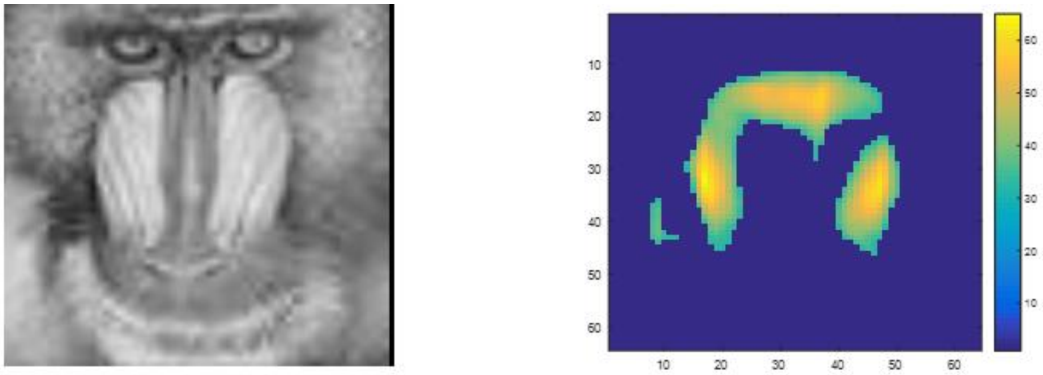


Figure 4. a) Complete Mandrill Image b) Mandrill's Signature

After applying the filter, all cells in the conflict matrix will have the conflicts for each cell in the dataset. After that. the matrix will be normalized using the maximum value. Thus, the matrix values will be [0 1]. In this case, the strongest conflicted cell(s) will have the value 1.

The next step is to select the threshold value. This threshold will be in the range [0 1], where 0 means that we are selecting the whole conflict matrix. Selecting the threshold value of 0.3 means that we are selecting the cells that have a conflict value of 0.3 or more. Figure 6 shows different signatures using different thresholds. As shown in Figure 5, higher threshold value means less details.

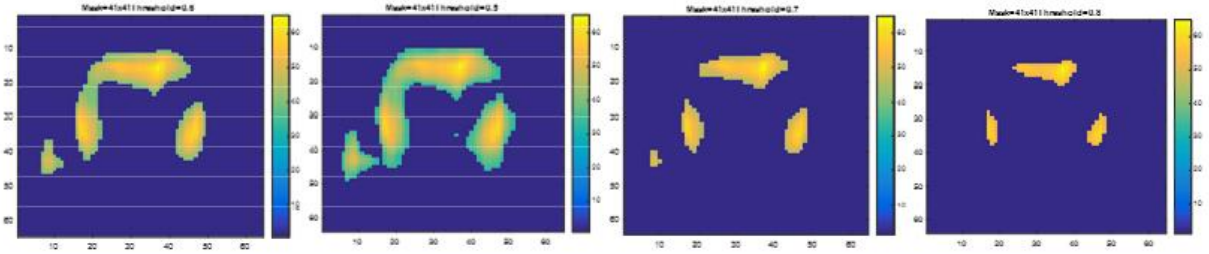


Figure 5. Signatures Using Threshold 0.5, 0.6, 0.7, and 0.8

If we have a reference dataset and aggregated versions of this reference dataset, then the same steps (applying the filter and then selecting a threshold value) will be applied on both the reference dataset and aggregated dataset. One question here is, what is the optimal size of the filter and the optimal value of the threshold. It is important to determine these values in order to minimize the error in the comparison. Our approach is able to detect the best filter size and the best threshold value and to do the correct mapping between the reference dataset and the aggregated dataset, which is very complicated to be done manually through human eyes.

After obtaining the optimal filter size and the optimal threshold value, the signature will be obtained from the reference image. By using the same filter and threshold value, the aggregated signature will be obtained. As the resulted datasets are normalized, the error or the difference between the two datasets can be calculated bitwise or segment wise in order to handle any transformation in the aggregated image. The error between datasets can be calculated using the following steps:

- Find the relative error using equation 4.
- Construct the percentile error for the resulted matrix.
- Find the area under the percentile error curve.
    - The higher the area means the bigger matching between the two images.

$$\text{Matching Error} = \frac{|\text{agg\_image\_value} - \text{orig\_image\_value}|}{(\text{orig\_image\_value} + 1)}$$

Equation 4

where agg_image_value is the pixel value of the aggregated image, and the org_image_value is the original pixel value.

Figure 6 shows the area under the percentile error curve using different thresholds and the bitwise comparison between the reference dataset and the aggregate dataset, where the x axis is the mask size and the y axis is the aggregate size. From this figure we can conclude that to get good results its recommended to use mask size equals or larger than aggregation size.

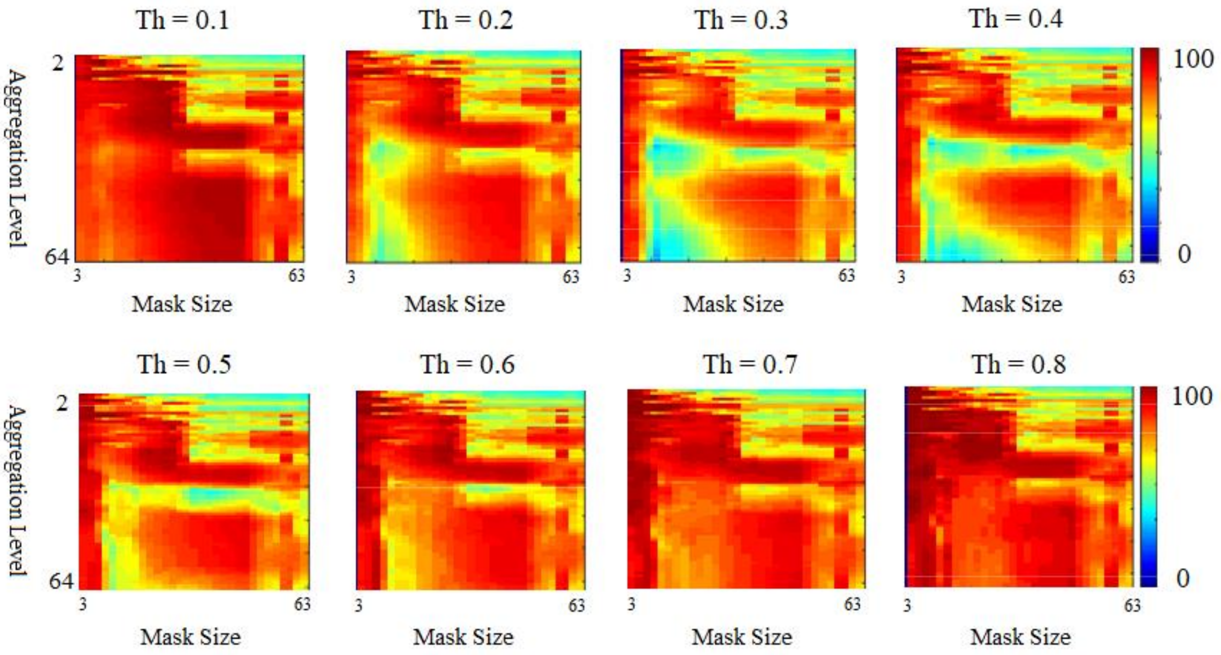


Figure 6. Area Under Percentile Error Curve Using Segmentation for Comparison

Figure 7 shows the area under the percentile error curve using different thresholds and the segmentation for the comparison between the reference dataset and the aggregate dataset, where x axis is the filter size, the y axis is the aggregate size. The color reflects the area under the percentile error curve, where blue means smaller area and red means larger area. It is clear the best results can be obtained using larger mask size and lower threshold.

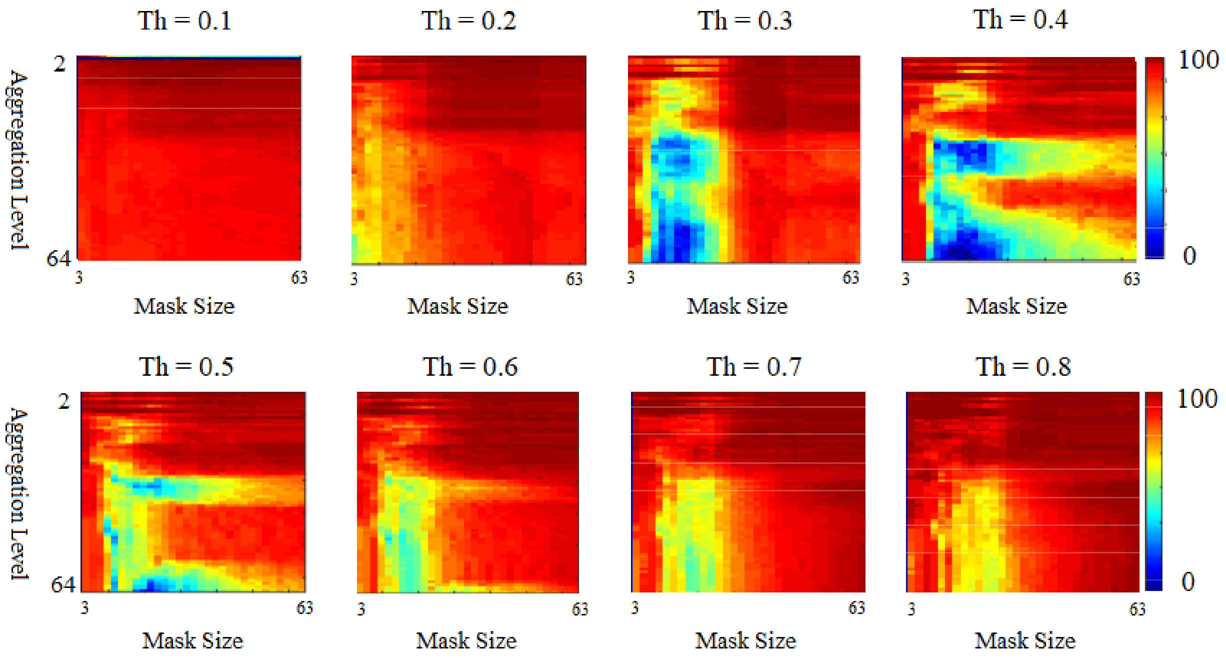
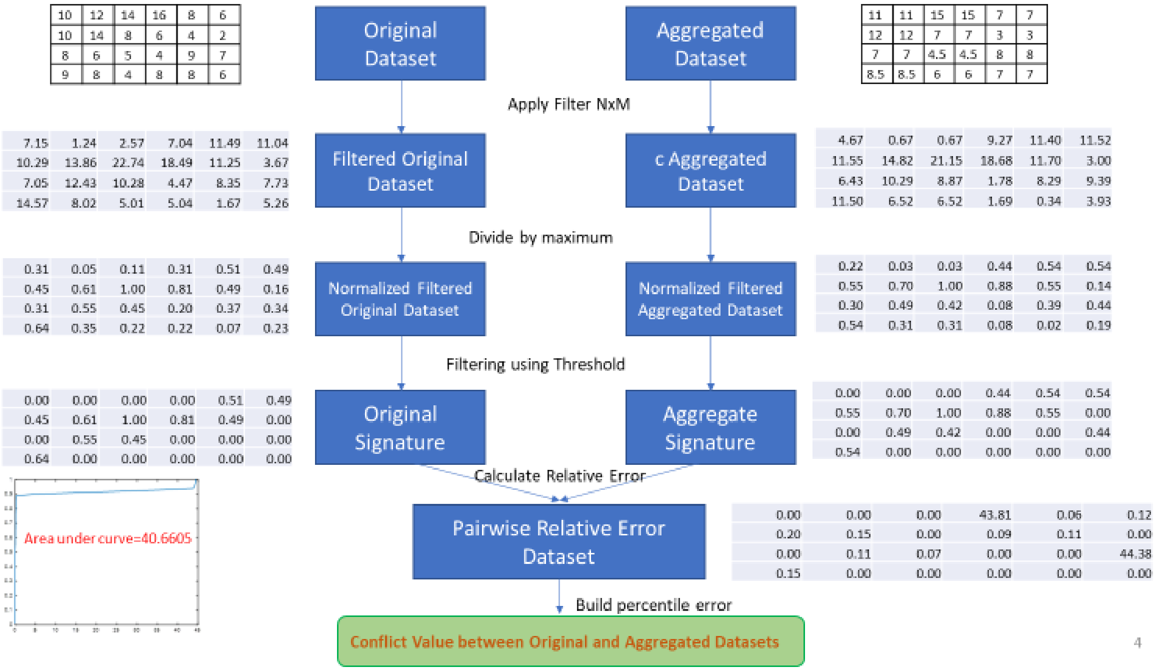


Figure 7. Area Under Percentile Error Curve Using Segmentation for Comparison

In order to select the filter size and threshold, first, we have to provide a balance between performance (the computation time and memory size) and the accuracy to be more accurate in the matching between datasets. If we are not concerned about the time and resources, we can choose a very low threshold such as, 0.1 and large filter size, since this low threshold includes fine grain data. If we are concerned about the computation time and memory, then we have to select large threshold and, in this case, we have coarser data. As the dataset is m x n and the filter is m' x n', then the computation cost is O(m*n*m'*n').

In general, larger filter size is better than smaller filter size. However, as we increase the filter size, the computation time and required memory will be larger. From figure 7 we can choose very low thresholds such as, 0.1 and 0.2 for higher accuracy. Large threshold values such as 0.7 and 0.8 provides better performance in terms of time and memory. It is also recommended to avoid midrange values such as, 0.3, 0.4, and 0.5. The reason is that when we move the threshold from 0.1 to 0.4, more details will be discarded, which affects the comparison between the aggregated images and the original images. This means that more zeros will be added for both of the conflict matrices that are related to aggregated image and original image. Thus, the distribution of relative errors will be changed and more non-zero values will appear. However, as we increase to higher threshold values such as 0.8, then more zeros will be added for both of the conflict matrices that are related to aggregated images and original image. As most of the values are zeros in both sides of the two conflict matrices, then the percentage of zero values of the relative error will be increased, but in this case, we take the most conflicting parts (informative parts of each image), which means that we increase the possibility of mismatching.

Figure 8 provides an example with real numbers to perform the comparisons between two datasets (original and aggregate version of the same dataset). In this figure, we have a subset of the original dataset in the left side and a subset of its corresponding aggregate dataset in the right side. We performed the aggregation using the average of each two adjacent cells in each row. We then applied our filter to find the conflict matrix for each dataset. We then normalized each conflict matrix by its maximum value. After that, we used a threshold value of 0.4 in order to filter the results, then we calculated the pair wise relative error and the final step was to build the percentile error.


Original Dataset
Aggregated Dataset
Apply Filter NxM
Filtered Original Dataset
c Aggregated Dataset
Divide by maximum
Normalized Filtered Original Dataset
Normalized Filtered Aggregated Dataset
Filtering using Threshold
Original Signature
Aggregate Signature
Calculate Relative Error
Pairwise Relative Error Dataset
Build percentile error
Conflict Value between Original and Aggregated Datasets
Area under curve=40.6605
4

10	12	14	16	8	6
10	14	8	6	4	2
8	6	5	4	9	7
9	8	4	8	8	6

11	11	15	15	7	7
12	12	7	7	3	3
7	7	4.5	4.5	8	8
8.5	8.5	6	6	7	7

7.15	1.24	2.57	7.04	11.49	11.04
10.29	13.86	22.74	18.49	11.25	3.67
7.05	12.43	10.28	4.47	8.35	7.73
14.57	8.02	5.01	5.04	1.67	5.26

4.67	0.67	0.67	9.27	11.40	11.52
11.55	14.82	21.15	18.68	11.70	3.00
6.43	10.29	8.87	1.78	8.29	9.39
11.50	6.52	6.52	1.69	0.34	3.93

0.31	0.05	0.11	0.31	0.51	0.49
0.45	0.61	1.00	0.81	0.49	0.16
0.31	0.55	0.45	0.20	0.37	0.34
0.64	0.35	0.22	0.22	0.07	0.23

0.22	0.03	0.03	0.44	0.54	0.54
0.55	0.70	1.00	0.88	0.55	0.14
0.30	0.49	0.42	0.08	0.39	0.44
0.54	0.31	0.31	0.08	0.02	0.19

0.00	0.00	0.00	0.00	0.51	0.49
0.45	0.61	1.00	0.81	0.49	0.00
0.00	0.55	0.45	0.00	0.00	0.00
0.64	0.00	0.00	0.00	0.00	0.00

0.00	0.00	0.00	0.44	0.54	0.54
0.55	0.70	1.00	0.88	0.55	0.00
0.00	0.49	0.42	0.00	0.00	0.44
0.54	0.00	0.00	0.00	0.00	0.00

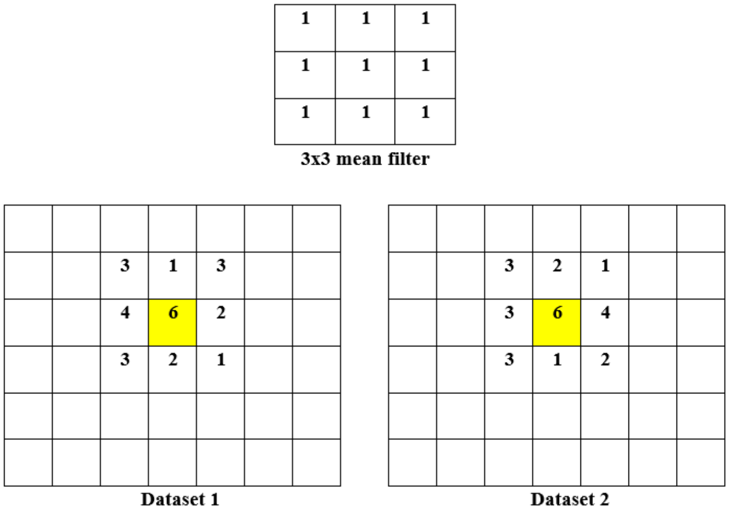
0.00	0.00	0.00	43.81	0.06	0.12
0.20	0.15	0.00	0.09	0.11	0.00
0.00	0.11	0.07	0.00	0.00	44.38
0.15	0.00	0.00	0.00	0.00	0.00

Figure 8. Similarity Assessment between two datasets

The previously mentioned filter does not work well when the mean of area to the right of each central cell equals to the mean of the left area of the central cell. Even though, the distribution of values maybe different in each area (left and right areas). The same limitation appears in the case of upper and lower areas. Therefore, we need to treat the two areas as different areas even if they have the same mean. Given the examples below, we can notice this limitation. Additionally, this limitation can appear when we use weighted mean filter, median, maximum, minimum, and Laplacian filter instead of the mean. For example, considering the following cases, we can see the limitation of each filter. Using these filters, we cannot always discriminate different datasets since we always get the same results as shown in the following examples.

We can see that the distribution of the data around the central cell is different in the two datasets (dataset1 and 2) as shown in Figure 9.

- By using a convolutional mean filter of size 3x3, the results from the two datasets are the same and equal to 2.78 and so we cannot discriminate the two datasets.
- By using a convolutional maximum filter of size 3x3, the results from the two datasets are the same and equal to 6. By using a convolutional minimum filter of size 3x3, the results from the two datasets are the same and equal to 1 and so we cannot discriminate the two datasets.
- By using a convolutional median filter of size 3x3 the results from the two datasets are the same and equals to 3 and so we cannot discriminate the two datasets.



1	1	1
1	1	1
1	1	1

3x3 mean filter

		3	1	3		
		4	6	2		
		3	2	1		

Dataset 1

		3	2	1		
		3	6	4		
		3	1	2		

Dataset 2

Figure 9. Mean Filter and 2 Datasets

- By using a convolutional weighted mean filter of size 3x3, the results from the two datasets (dataset 3 and 4) are the same and equal to 3.56 using the following weighted 3x3 filter, as shown in Figure 10.

1	1	1
1	2	1
1	1	1

3x3 weighted mean filter

		6	3	3		
		2	3	4		
		5	1	2		

Dataset 3

		4	2	5		
		3	2	6		
		3	1	4		

Dataset 4

Figure 10. Weighted Mean Filter and 2 Datasets

- By using a Laplacian filter of size 3x3, the results from the two datasets (dataset 5 and 6) are the same and equal to 1 using the weighted 3x3 filter as shown in Figure 11 and so we cannot discriminate the two datasets.

1	1	1
1	-8	1
1	1	1

3x3 Laplacian filter

		1	2	6		
		2	3	2		
		5	4	3		

Dataset 5

		2	4	3		
		1	2	2		
		2	2	1		

Dataset 6

Figure 11. Laplacian Filter and 2 Datasets

- By using an edge detection filter of size 5x5 to detect the vertical edges, the results from the two datasets (dataset 7 and 8) are the same and equal to 0 using the following 5x5 filter, as shown in Figure 12 and so we cannot discriminate the two datasets.

-1	-1	0	1	1
-1	-1	0	1	1
-1	-1	0	1	1
-1	-1	0	1	1
-1	-1	0	1	1

5x5 filter to detect vertical edges

	4	4	2	4	6	
	4	3	1	2	3	
	4	4	3	5	2	
	4	3	1	5	4	
	4	2	2	2	3	

Dataset 7

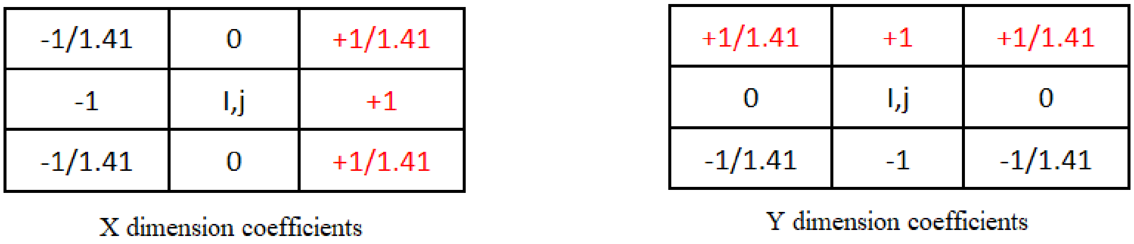
	6	1	2	4	4	
	4	2	1	3	3	
	2	5	3	6	1	
	4	2	1	4	2	
	6	2	2	2	5	

Dataset 8

Figure 12. Edge Detection filter and 2 Datasets

3.2.2 Signature Filter Design

Given the previously mentioned limitations for the different filters, we need to design a more efficient filter that can assign a unique value for the changes around the central cell for each different distribution of data around the central cell. In our approach, we used a scanning filter to estimate the changes around each data cell. The filter size is n x m. In our experiments, we used the same values for m and n. This means that the filter is n x n. The filter coefficients have different values according to the Euclidean distance from the filter center. The filter has two components in the x and y dimensions as shown in figure 13.



-1/1.41	0	+1/1.41
-1	I,j	+1
-1/1.41	0	+1/1.41

X dimension coefficients

+1/1.41	+1	+1/1.41
0	I,j	0
-1/1.41	-1	-1/1.41

Y dimension coefficients

Figure 13. X and Y coefficients for filter of size 3 x 3

Using this filter, we still may have the limitation of previous filters, so we will divide the area around the central cell to up, down, right, and left areas. Then we will divide each area into different partitions and aggregate each partition and assign different weights for each partition according to the distance between this partition and the central cell.

We used the equations below to find the final difference in both x and y directions for each cell. Table 2 provides a description of the symbols use in each equation.

Table 2. Signature sub equations

Symbol	Description
mqR	The weighted average for the q part in the right side of the central cell
mqL	The weighted average for the q part of the left side of the central cell
mqU	The weighted average for the q part of the upper side of the central cell
mqD	The weighted average of the q part of the down side of the central cell
diffq_x	The absolute difference between the right and left weighted average of the q part
diffq_y	the absolute difference between the up and down weighted average of the q part
G	The number of parts to the right or left sides of the central cell
D	The number of parts to up or down sides of the central cell
Diff_x	The average of the differences in the x direction
Diff_y	The average of the differences in the y direction
Diff(i,j)	The net difference around the central cell (i,j)
Signature(i,j)	The relative net differences around the central cell.
P and Z	To control the portions of signature and original respectively.

$$mqR = \frac{1}{Agg_x(2n_y+1)} \sum_{k=i-n_y}^{i+n_y} \sum_{z=j+(q-1)Agg_x+1}^{j+q\,Agg_x} \frac{value(k,z)}{dist((i,j),(k,z))}$$

$$mqL = \frac{1}{Agg_x(2n_y+1)} \sum_{k=i-n_v}^{i+n_y} \sum_{z=j+(q-1)Agg_x+1}^{j-(q-1)Agg_x-1} \frac{value(k,z)}{dist((i,j),(k,z))}$$

$$mqU = \frac{1}{Agg_y(2n_y+1)} \sum_{z=j-n_x}^{j+n_x} \sum_{k=i-q\,Agg_y}^{i-(q-1)Agg_y-1} \frac{value(k,z)}{dist((i,j),(k,z))}$$

$$mqD = \frac{1}{Agg_y(2n_y+1)} \sum_{z=j-n_x}^{j+n_x} \sum_{k=i+(q-1)Agg_y+1}^{i+q\,Agg_y} \frac{value(k,z)}{dist((i,j),(k,z))}$$

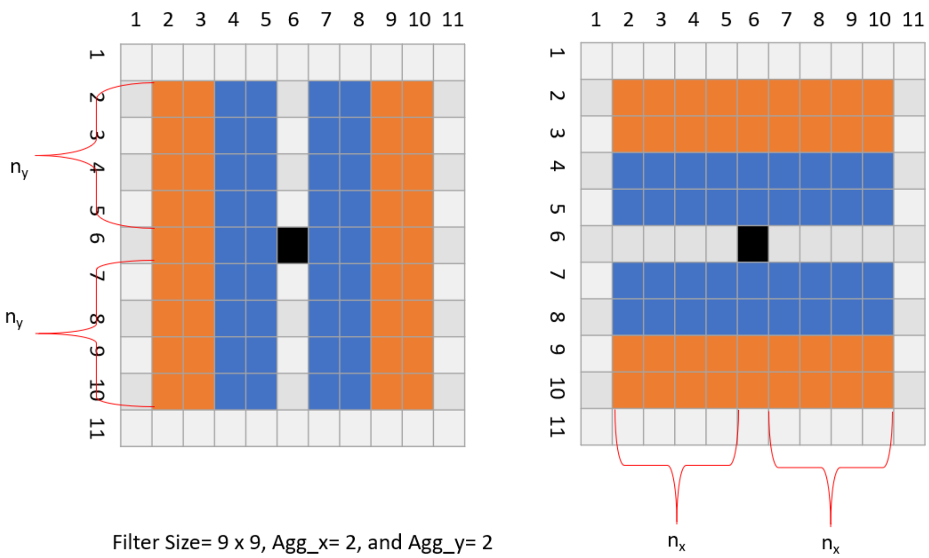
$$\mathrm{diffq\_x} = |\ \mathrm{mqR} - \mathrm{mqL}\ | \qquad \mathrm{Diff\_x} = \frac{1}{G}\sum_{q=1}^{G} \mathrm{diffq\_x}$$

$$\mathrm{diffq\_y} = |\ \mathrm{mqU} - \mathrm{mqD}\ | \qquad \mathrm{Diff\_y} = \frac{1}{D}\sum_{q=1}^{D} \mathrm{diffq\_y}$$

$$\mathrm{Diff}\,(i,j) = \sqrt{(\,\mathrm{Diff\_x}\,)^2 + (\,\mathrm{Diff\_y}\,)^2}$$

$$\mathrm{Signature}\,(i,j) = (\,1 + (\,1 - e^{-(a)\mathrm{Diff}(i,j)}\,)^{p}\,)(\,1 + m(i,j)\,)^{z}$$

In figure 14, the filter size in both directions is 9, the filter x dimension is in the form of ($2n_x + 1$) and y dimension is in the form ($2n_y + 1$). There are four columns in both right and left sides of the central cell (the black one in figure 15). Additionally, there are four rows in both top and bottom of the central cell. The aggregation level = 2 in both directions (x and y), which means that we are aggregating each two adjacent columns in the x direction. It also means that we are aggregating each two adjacent rows in the y direction. As shown in figure 15, we have four rows above the central cell. The colors reflect the aggregated rows. Rows with numbers 4 and 5 are aggregated together. Rows with numbers 2 and 3 are also aggregated together. Furthermore, the columns with numbers 4 and 5 are aggregated together, as well as columns with numbers 2 and 3 are aggregated together. The aggregation is done by taking the weighted average. Cells near the central cell have higher weights. We used the Euclidean distance to control the weight of each cell.



Filter Size= 9 x 9, Agg_x= 2, and Agg_y= 2

Figure 14. Filter of Size 9 x 9 in both X and Y Directions

In our approach, we depend on the symmetry, i.e., the left side and the right side of the central cell have the same size, and on the distance to the central cell. The blue parts in the figure 16 have the same size and the same distance to the central cell. This is also applied to the top and bottom sides. To estimate the changes around the central cell in the x direction, we compare the difference between left and right sides. In the following example, we compare the aggregated columns (4,5) with the aggregated columns (7, 8) as shown in figure 15. We also compare the aggregated columns (2,3) with the aggregated columns (9, 10). Therefore, we have two results of the comparisons in the x direction (diff1_x and diff2_x). On the other hand, to estimate the changes around the central cell in the y direction, we compare the difference between the top and bottom sides. In this example, we compare the aggregated rows (4,5) with the aggregated rows (7,8). We also compare the aggregated rows (2,3) with the aggregated rows (9,10), and thus, we have two results of the comparisons in the y direction (diff1_y and diff2_y).

$$\text{Diff\_x} = (\text{diff1\_x} + \text{diff2\_x})/2$$

$$\text{diff1\_x} = |\,\text{m1R} - \text{m1L}\,|$$

$$\text{diff2\_x} = |\,\text{m2R} - \text{m2L}\,|$$

$$\text{m1R} = \frac{1}{(2(4)+1)\,2}\sum_{k=2}^{10}\sum_{z=7}^{8}\frac{\text{Value }(k,z)}{\text{dist}((i,j),(k,z))}$$

$$\text{m1L} = \frac{1}{(2(4)+1)\,2}\sum_{k=2}^{10}\sum_{z=4}^{5}\frac{\text{Value }(k,z)}{\text{dist}((i,j),(k,z))}$$

$$\text{m2R} = \frac{1}{(2(4)+1)\,2}\sum_{k=2}^{10}\sum_{z=9}^{10}\frac{\text{Value }(k,z)}{\text{dist}((i,j),(k,z))}$$

$$\text{m2L} = \frac{1}{(2(4)+1)\,2}\sum_{k=2}^{10}\sum_{z=2}^{3}\frac{\text{Value }(k,z)}{\text{dist}((i,j),(k,z))}$$

$$\text{Diff\_y} = (\text{diff1\_y} + \text{diff2\_y})/2$$

$$\text{diff1\_y} = |\,\text{m1U} - \text{m1D}\,|$$

$$\text{diff2\_y} = |\,\text{m2U} - \text{m2D}\,|$$

$$\text{m1U} = \frac{1}{2\,(2(4)+1)}\sum_{k=4}^{5}\sum_{z=2}^{10}\frac{\text{Value }(k,z)}{\text{dist}((i,j),(k,z))}$$

$$\text{m1D} = \frac{1}{2\,(2(4)+1)}\sum_{k=7}^{8}\sum_{z=2}^{10}\frac{\text{Value }(k,z)}{\text{dist}((i,j),(k,z))}$$

$$\text{m2U} = \frac{1}{2\,(2(4)+1)}\sum_{k=2}^{3}\sum_{z=2}^{10}\frac{\text{Value }(k,z)}{\text{dist}((i,j),(k,z))}$$

$$\text{m2D} = \frac{1}{2\,(2(4)+1)}\sum_{k=9}^{10}\sum_{z=2}^{10}\frac{\text{Value }(k,z)}{\text{dist}((i,j),(k,z))}$$

Figure 15 provides another detailed example that shows a part of a dataset with a scanning filter of size 7 X 7. The scanning filter has two dimensions. The aggregation level is 1. The x dimension is in the form ($2n_x + 1$) and the y dimension is in the form ($2n_y +1$). Therefore, in the example shown in figure 15, $n_x=3$ means that there are three columns to the right of the central cell and three columns to the left. Additionally, $n_y=3$ means that there are three rows above the central cell and three rows below the central cell.

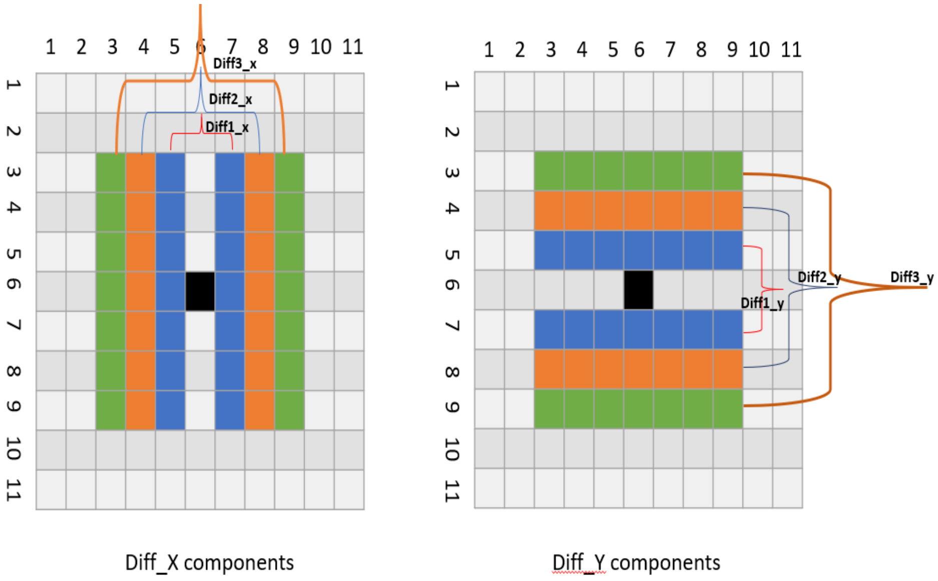
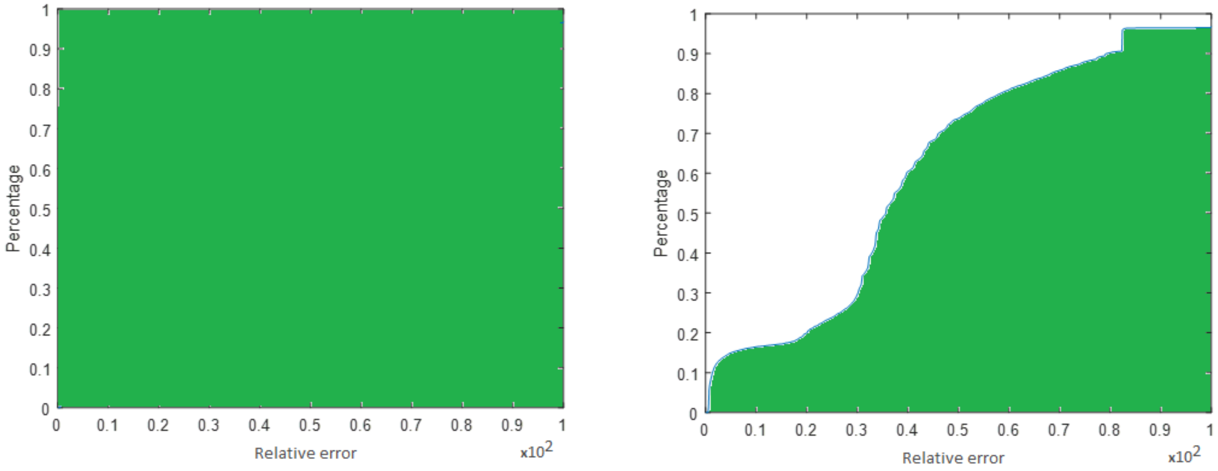


Figure 15. Filter of Size 7 x 7 in Both X and Y Directions

3.2.3 Experimental Study: Objects Comparison Using Signatures:

In order to show the effectiveness of our signature we used images (Mandrill, Laura and Tipper). We created different aggregated versions for each reference (original image), then we used our signature with different filter sizes such as 10, 16, and 61. Tables 3-6 show the total relative error between each reference image and the aggregated version. For example, aggregation 16 means that we need to aggregate the first 16 cells in the row and assign the mean value to all the 16 cells, and then aggregate the second 16 cells (starting from cell 17 to cell 32) and repeat this process for all cells in each row. The values shown in the tables below represent the difference between the area under the curve for the relative error (when all error values equal zeros) and the area under the curve as shown in Figure 16. When the difference converges to zero, then the two datasets that we want to compare become close to each other (belongs to the same object).



(a) **(b)**

Figure 16. Percentile error when all errors=0 and when errors have non-zero values

From Figure 16, we can notice the area of the white region equals difference between the area under the curve when all values of errors and the area under the curve when errors have non-zero values. The values in tables 3, 4, and 5 represent the area of white region in Figure 16. As this value converge to zero it means the percentage of errors with value zero is higher and then the two objects, we are comparing are closer to each other.

For example, as shown in Table 5, when we compare an aggregated version of Mandrill dataset with an original version of Mandrill, then the difference is 29.1. however, the difference between the aggregated version of Mandrill and the original version of Laura and Tipper is 47.05 and 94.81 respectively. The same thing applies to tables 3, 4, 5 and 6. The best results can be achieved when the filter size is greater than the aggregate level as shown in table 5.

Table 3. Matching relative errors for aggregate level= 16, Filter= 16

	Original		
Aggregated	Mandrill	Laura	Tipper
Mandrill	56.85	83.30	72.73
Laura	80.67	65.68	71.04
Tipper	72.8	67.0	61.06

Table 4. Matching relative errors for aggregate level = 32, Filter= 10

	Original		
Aggregated	Mandrill	Laura	Tipper
Mandrill	88.91	87.94	90.72
Laura	92.73	86.62	92.61
Tipper	95.60	95.17	45.61

Table 5. Matching relative errors for aggregate level = 12, Filter= 61

	Original		
Aggregated	Mandrill	Laura	Tipper
Mandrill	29.1	47.05	94.81
Laura	81.21	25.71	93.39
Tipper	84.25	78.83	29.13

Table 6. Matching relative errors for aggregate level = 60, Filter= 61

	Original		
Aggregated	Mandrill	Laura	Tipper
Mandrill	48.14	66.54	93.55
Laura	93.87	34.81	84.30
Tipper	62.4	61.30	57.34

From the above tables, we can conclude that more accurate results can be obtained when the filter size is greater than the aggregation level. For example, when the filter size is 61 and the aggregate level is 12, then the diagonal will contain the smallest matching error values as compared to the other values and the variance will be high.

In the next experiment, we compared six images (primarily related to three persons in different poses). Table 7 shows results of comparison using our approach. L1 refers to person 1 who looks to the left. R1 refers to person 1 who looks to the right, and so on as shown in Figure 17. The comparison was done using the difference between the area under the curve for each image. Therefore, minimum value means higher similarity between two datasets (images). The results show that the comparison between L1 and R1 is the minimum difference value when L1 is compared with R2 and R3. Additionally, the comparison between L2 and R2 is the minimum difference value when L2 is compared with R1 and R3, and the same thing applies to L3. Table 8 shows results of comparison using original dataset (images). The comparison between L1 and R1 is the maximum value of comparison. The comparison between L1 and R3 is the minimum value of comparison, which means that L1 is highly similar to R3, instead of R1. The same thing applies to L2 and L3. Thus, we can conclude that our approach can transform the data to be more informative and thus get higher accuracy of comparison between datasets.

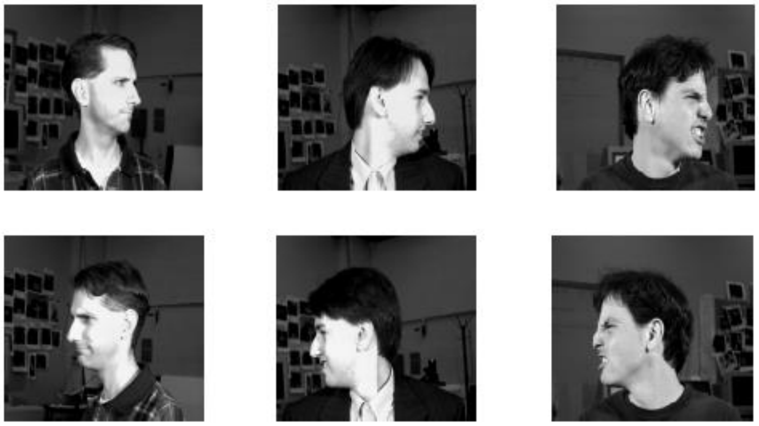
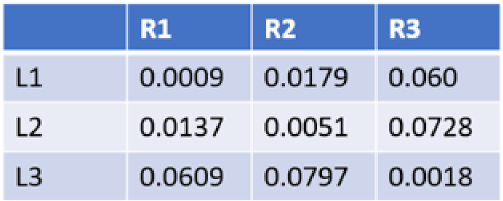


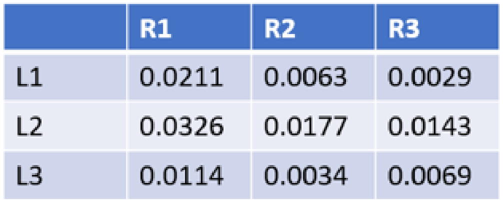
Figure 17. Three Different Persons with Different Poses

Table 7. Differences between images using our signature



	R1	R2	R3
L1	0.0009	0.0179	0.060
L2	0.0137	0.0051	0.0728
L3	0.0609	0.0797	0.0018

Table 8. Differences between images using original version



	R1	R2	R3
L1	0.0211	0.0063	0.0029
L2	0.0326	0.0177	0.0143
L3	0.0114	0.0034	0.0069

Figure 18 shows the area under the curve for L1 and R2 respectively using our approach. In order to find the area under the curve, we normalized the dataset through the division by the maximum value in the dataset. Therefore, the values of the dataset will be between 0 and 1. Then we divided this range using a step value such as 0.001 and thus the x axis represents 1000 points such as, 0.001, 0.002, 0.003 and so on. The y axis represents the count of all values that are less than or equal the x value. Y (0.002) = count of all data values in the normalized dataset that equal to 0.002 or less.

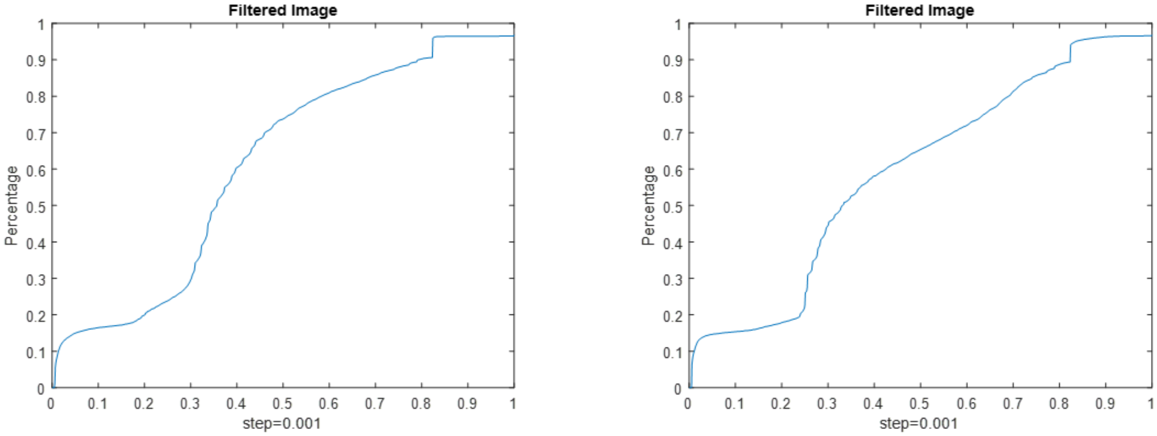


Figure 18. Area Under Curve for L1 and R2 Respectively Using Our Approach

From Figure 18, there is a noticeable difference in the area under the curve for L1 and R2. However, in Figure 19, we can see that using the original data to calculate the area under the curve does not provide a noticeable difference in the area under the curve for L1 and R2.

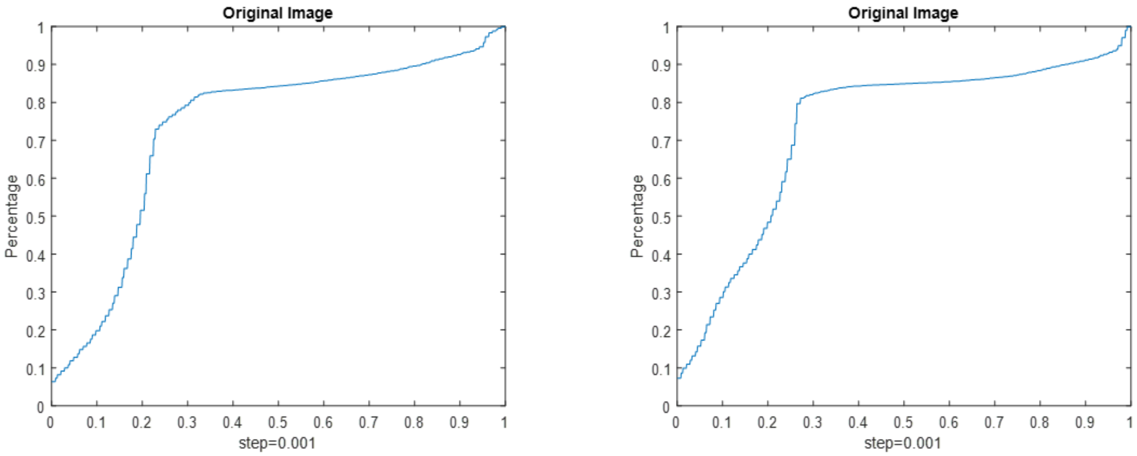


Figure 19. Area Under Curve for L1 and R2 Respectively Using Original Datasets

For the previous experiments we can notice the importance of our signature in distinguishing different objects as compared to the use of the original data only.

Finally, and based on our experimental study, we observed that the signature can preserve the information in the aggregated dataset and thus make it more informative.

3.3 RESEARCH QUESTION 2: HOW TO RELATE THE INFORMATION AT DIFFERENT AGGREGATION LEVELS AND HOW TO BUILD AN EFFICIENT RETRIEVAL ARCHITECTURE FOR AGGREGATED DATASETS?

3.3.1 Aggregated Information Retrieval Approach (sigMatch)

After creating our signature, we need to explore the efficiency of this signature in retrieving and indexing aggregated images using different aggregation levels. For this purpose, we built an architecture of the aggregated information retrieval approach using signature space and original space. We are calling this approach SigMatch.

Given a raw dataset that is rich in details and a repository of aggregated datasets that lacks some details, what is the relationship between the detailed dataset and each dataset in the repository? Comparing detailed dataset and aggregated dataset is challenging and will not provide accurate results. In order to solve this challenge, we could develop a retrieval approach that is based on the signature space instead of the original space. Since we can highlight the most informative parts in each dataset using the signature space, we can reduce the distance between the detailed dataset and the corresponding aggregated dataset, which in turn improves the retrieval process. We tested our approach using different detailed datasets and aggregated datasets. The result showed a considerable improvement in the accuracy of indexing process.

We addressed this research question by developing an aggregated information retrieval approach using the signature as mentioned earlier in research question 1. As shown in Figure 20, we have an aggregated repository of different levels of aggregation and we also have detailed images in the left side, we need to relate each detailed image with its corresponding image(s) in the aggregated repository. This object comparison task can be done in two ways (as shown in Figure 21): (1) using original data and aggregated data only or (2) using the signatures of detailed images and aggregated images in the repository. In the first way, each image in the set of query images will be compared with each image in the images repository and thus there will be an accuracy array for each image in the query images. Accuracy array of index i includes multiple results of comparing image i with repository images. The process of comparing will be applied for all other images in the set of query images. In the second way, we used signature of images instead of original images. So, the accuracy array of index i includes multiple results of comparing signature of image i with signatures of the images in the repository. The comparing is done using structural similarity index measure (SSIM) since it is an improvement of the traditional methods such as mean squared error (MSE) and peak signal to noise ratio (PSNR) [41].

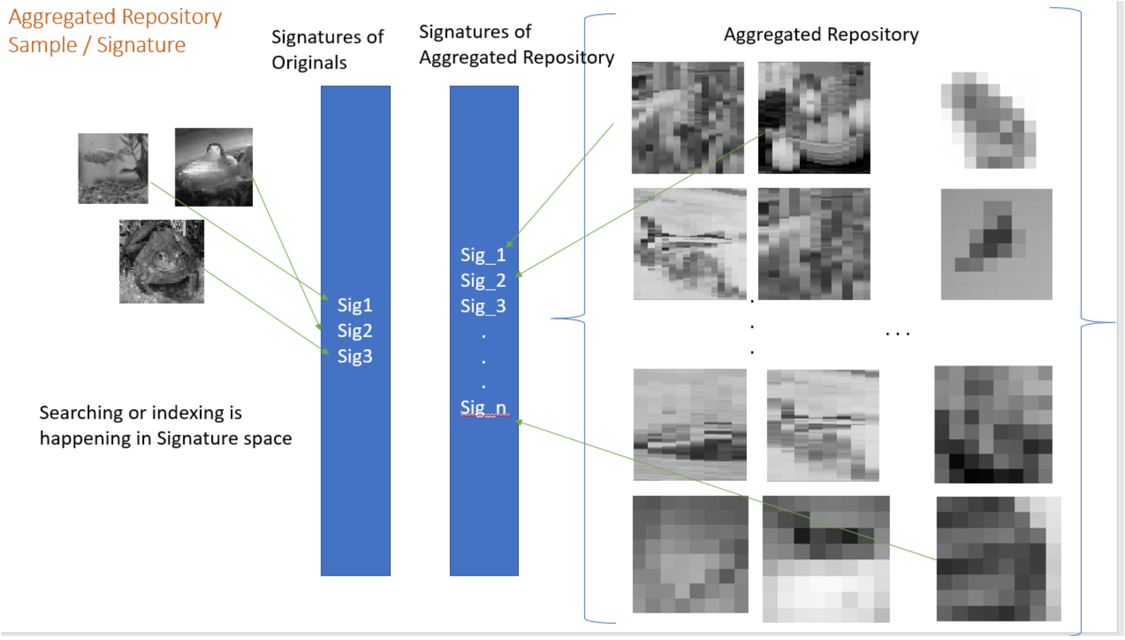


Figure 20: Example of Aggregated Repository

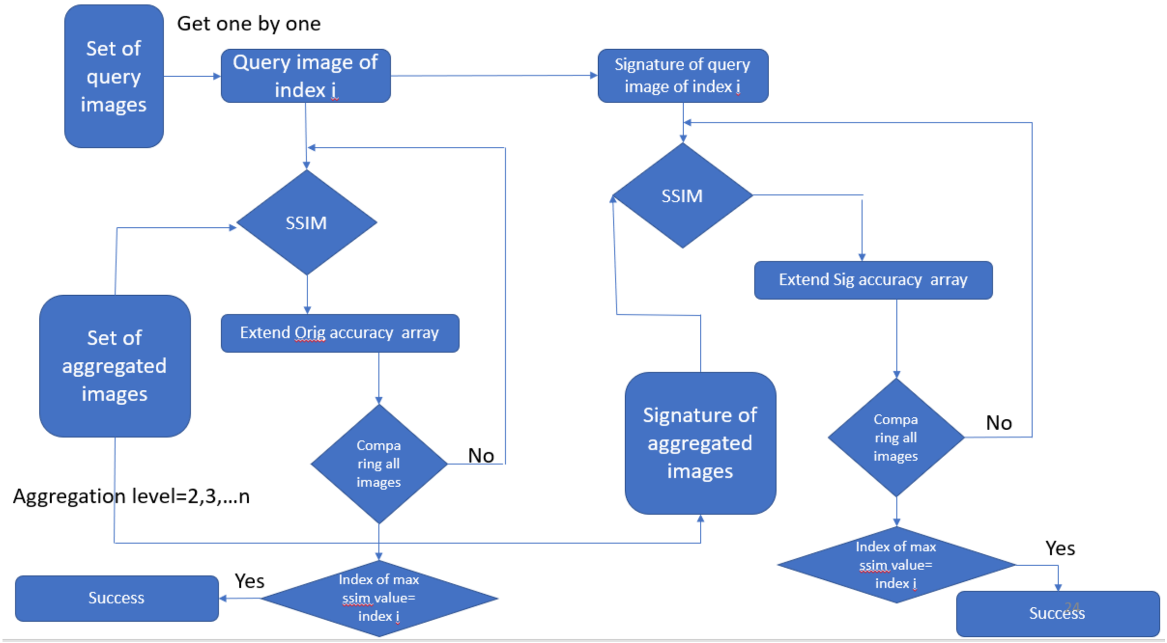

Set of query images
Get one by one
Query image of index i
Signature of query image of index i
SSIM
SSIM
Extend Sig accuracy array
Set of aggregated images
Extend Orig accuracy array
Signature of aggregated images
Compa ring all images
No
Compa ring all images
No
Aggregation level=2,3,…n
Index of max ssim value= index i
Yes
Success
Yes
Index of max ssim value= index i
Success

Figure 21: Object Comparisons Approaches

We tested our approach using the CIFAR-10 dataset [42] that consists of 60000 32x32 colored images and divided into 10 classes with 6000 images per class as shown in Figure 22. We created an aggregated repository of this dataset, the aggregation was done at different levels such as, 2, 4, 6, 8, and 10. The aggregation level represents the number of the adjacent cells that are aggregated together. For example, aggregation level of 4 indicates that we are aggregating each adjacent 4 cells. This means that cells of index 1,2,3 and 4 are aggregated together and cells of index 5, 6, 7 and 8 are aggregated together. In our experiment, we created a dataset that consists of 5 categories and each category has 500 images. In this experiment, we assessed the similarity or the distance between each image and the corresponding image in the aggregated repository using the original and signature space.

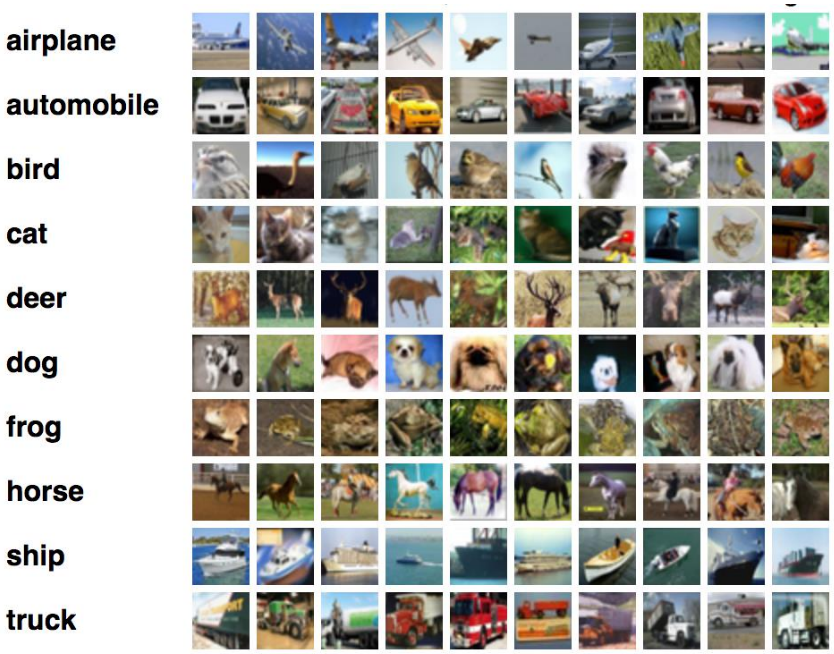


Figure 22: CIFAR-10 Sample Images

Source: https://becominghuman.ai/training-mxnet-part-2-cifar-10-c7b0b729c33c

Figure 23 shows how we measure the distances between original image and different aggregated versions of the image in the original and signature spaces. We created a signature for each original and aggregated image. We could measure the distance between an image and aggregated image using any aggregation level through the measure of distance between original and the aggregated image, or through the measure of distance between the signature of the original image and the signature of the aggregated image.

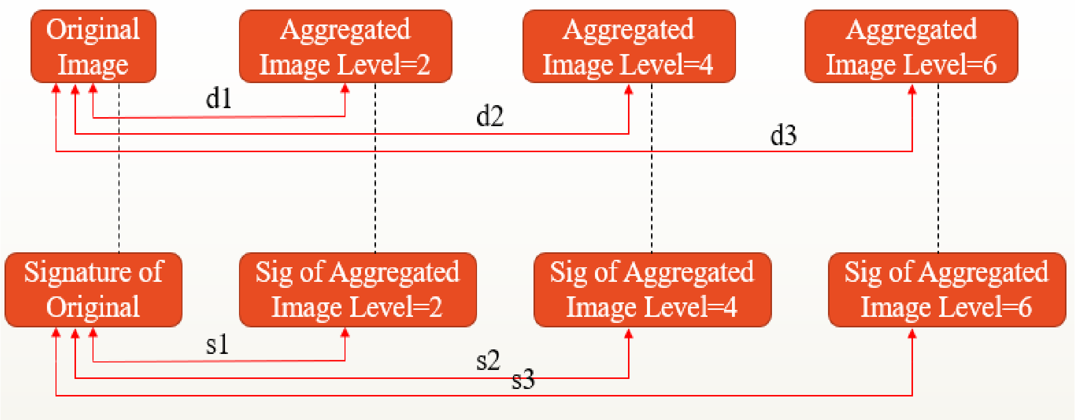


Figure 23. Distances in Original Space and Signature Space

Here:

d1 is the distance between the original image and aggregated form at level 2

d2 is the distance between the original image and aggregated form at level 4

s1 is the distance between the signature of the original image and the signature of the aggregated form at level 2

s2 is the distance between the signature of the original image and the signature of the aggregated form at level 4.

The results of comparison are shown in Figure 24, we observe that the average of distances using signatures is much smaller than the average of distances using original. This means that our approach can detect the related images using signatures more precisely than using original images and aggregated images.

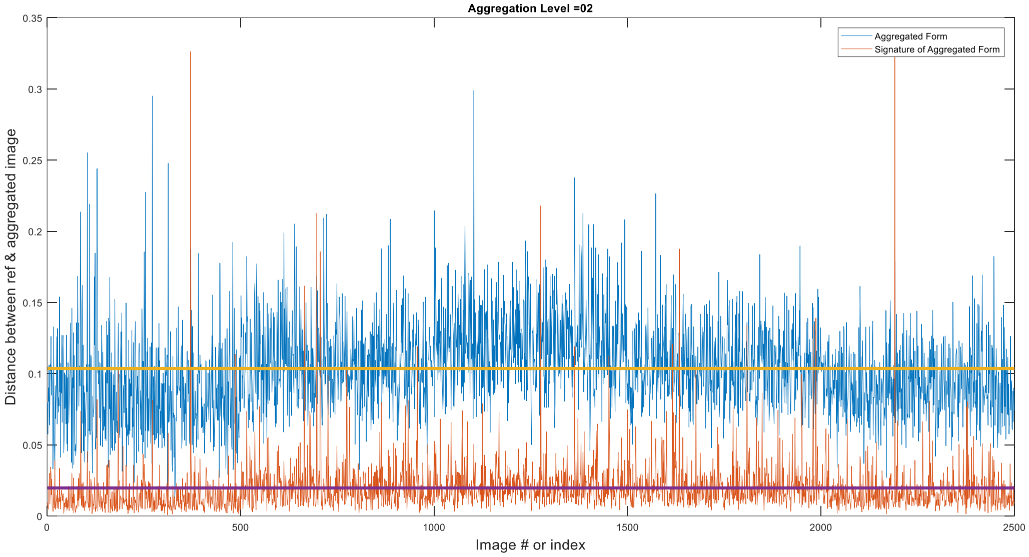


Figure 24. Distance Between Original and Aggregated Images in Original Space and Signature Space

The percentile of the distances using original and signatures is shown in Figure 25. As shown in the figure, we can see that the area under the curve can be used to reflect the distribution of distances. It is clear that larger area under the curve indicates smaller distances. It is also clear that the area the under curve for the signature curve is larger than the area under the curve for the original curve. This means that the distances using signature is smaller than distances using original.

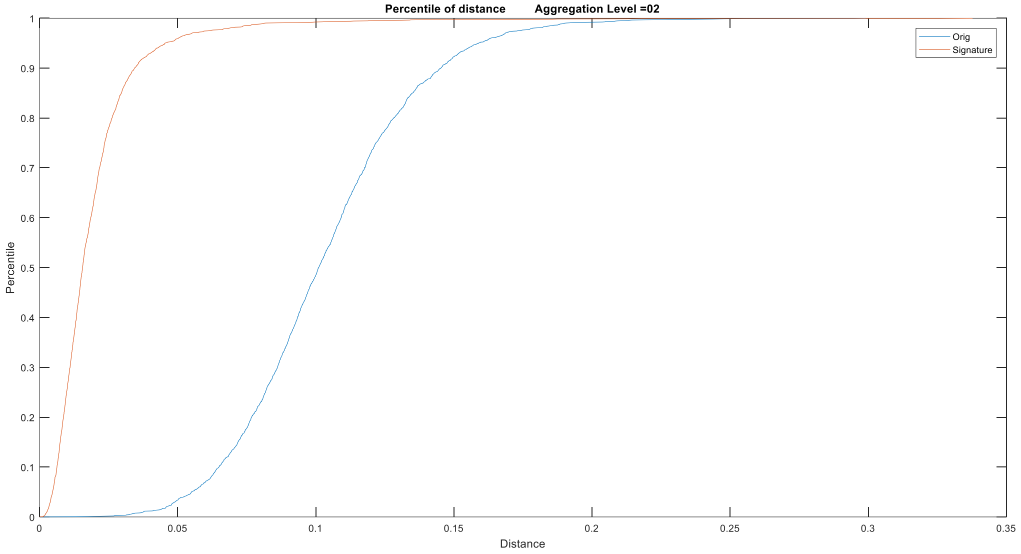


Figure 25. Percentile of Distances for Original and Signature

We repeated this procedure using different aggregation levels including, 4,6,8, and 10. Figure 26 shows the area under the curve. As shown in this figure, the area under the curve for each aggregation level using signature space is greater than the area under the curve using original space. The average of difference between the two areas under the curve using originals and signatures equals 0.3.

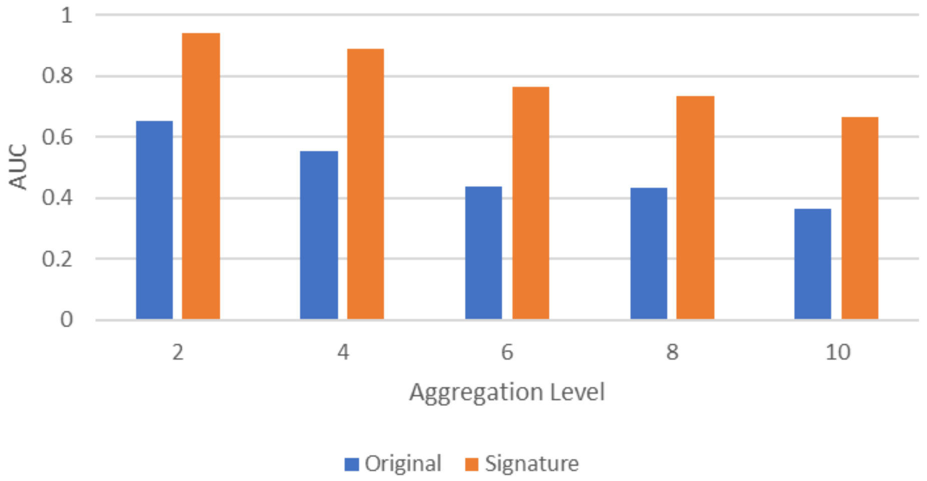


Figure 26. Area Under the Curve- Same Category

After that, we performed the comparison between different categories as shown in table 9. In this case, each image is not compared with the corresponding aggregated image. For example, the original airplane image is compared with aggregated frog image and the original horse image is compared with aggregated truck image.

Table 9. Categories for Comparison

Original Category	**Aggregated Category**
Airplane	Frog
Cat	Airplane
Frog	Cat
Horse	Truck
Truck	Horse

Figure 27 shows the area under the curve for distances using original space and signature space within different categories. In this figure, we can see that there is no big difference between the distances using original space and signature space. The average of difference between the two areas under the curve using originals and signatures equals 0.15.

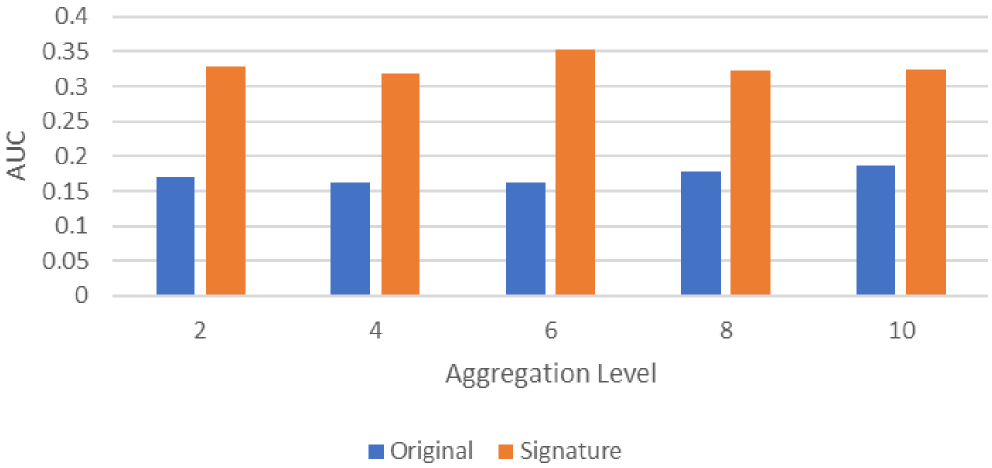


Figure 27. Area Under the Curve- Different Categories

Figure 28 shows the area under the curve for the range between similarity and non-similarity areas under the curve. Large difference indicates more precise decision can be done. Our approach provides a wider range, which means more accurate decision in comparing two objects.

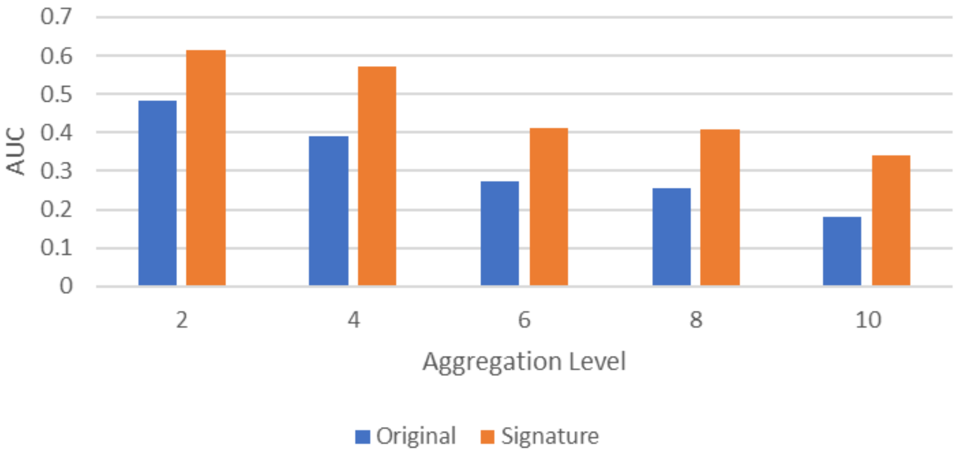


Figure 28. Area Under the Curve for the Range Between Similarity and Non-similarity

It is clear that for all aggregation levels, the range between similarity region and non-similarity region is wider in the case of signature than the range in the case of original. The decision regions are shown in figure 29. When the range is narrow as in the original, the decision of similarity and non-similarity will be more difficult, and the error rate will increase.

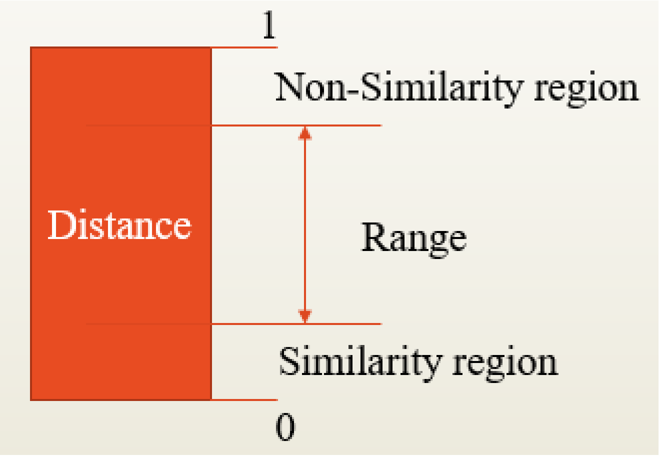


Figure 29. Similarity and Non-Similarity Regions

3.3.2 Performance of Signature Based Retrieval

In this experiment, we looked for the most corresponding image(s) from the aggregated repository. We executed a query as shown below:

Query: Given an image A, retrieve the top k corresponding images from the aggregated repository.

Figure 30 shows different aggregated datasets and original datasets. In this experiment, for each image in the original dataset, the corresponding image(s) should be retrieved for the corresponding aggregated data repository. We used two methods for retrieving the corresponding images from the aggregated repositories including, using original space and using signature space. In the signature space method, we create a signature for the original and each object in the aggregated repository.

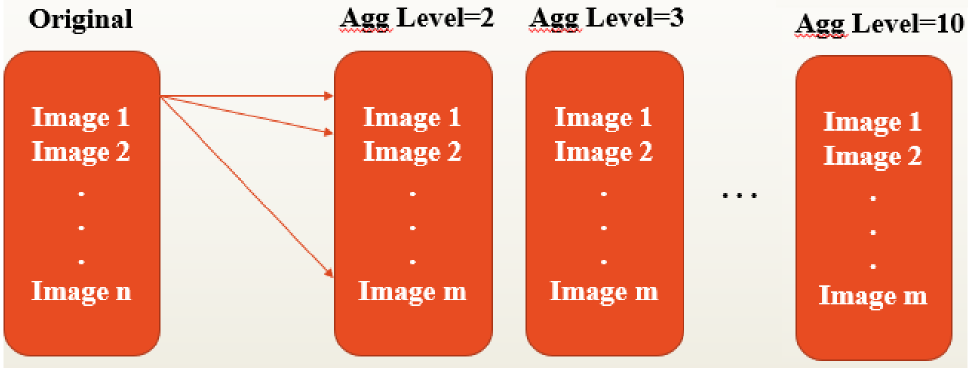


Figure 30. Retrieval from Aggregated Repository

In the first method of retrieval, we perform the retrieval from the aggregated repository. In the second method, we create a signature for each object or image in the aggregated repository. In both methods, we use the structural similarity index measure (SSIM) to measure the similarity between any two images. We then measure the TP, FP, and Precession for the retrieval process using original and signatures. The net TP, FP, and Precision for N images are calculated using the following equations:

$TP=mean\ (TP_1, TP_2, TP_3, . . . ,TP_n)$

$FP=mean\ (FP_1, FP_2, FP_3, . . . ,FP_n)$

$Precision=mean\ (P_1, P_2, P_3, . . . ,P_n)$

Figure 31 shows the average TP using different top K, where TPi equals 1 when the SSIM of the image and the aggregated image is among the Top K results. We can notice that the average of TP using signatures is greater than the average of TP using originals for all aggregation levels.

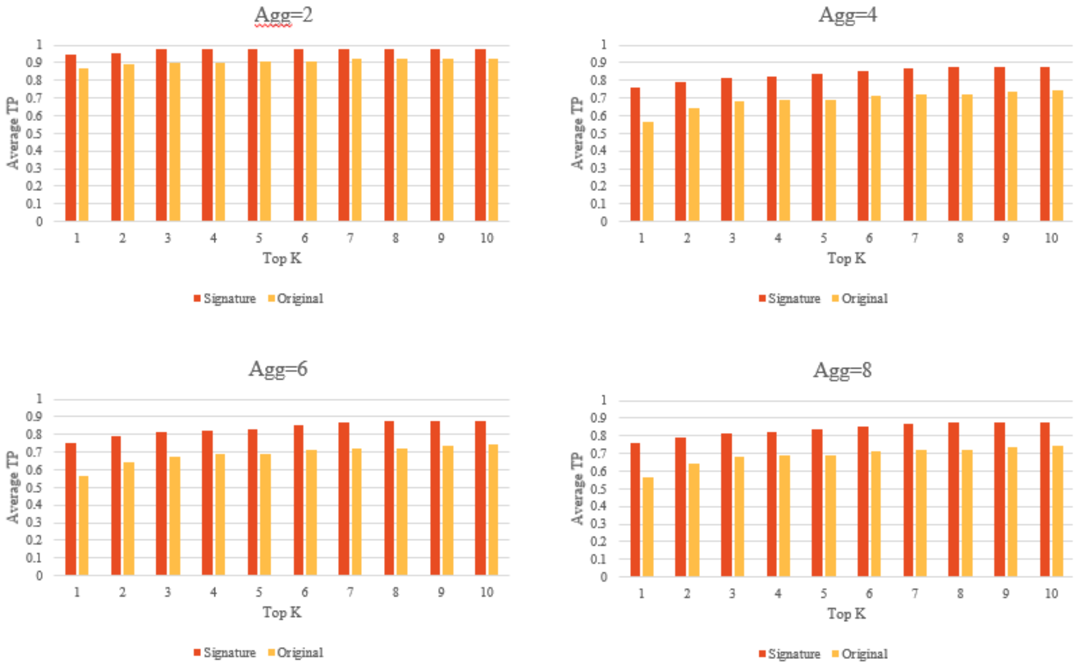


Figure 31. Average of True Positive Values

The average of FP using different aggregation levels are shown in figure 32. We can notice that as we increase the value of K the average of FP increases also we can notice that if we want to be more precise then we have to set K=1 and in this case the average of FP using signatures is less than the average of FP using originals.

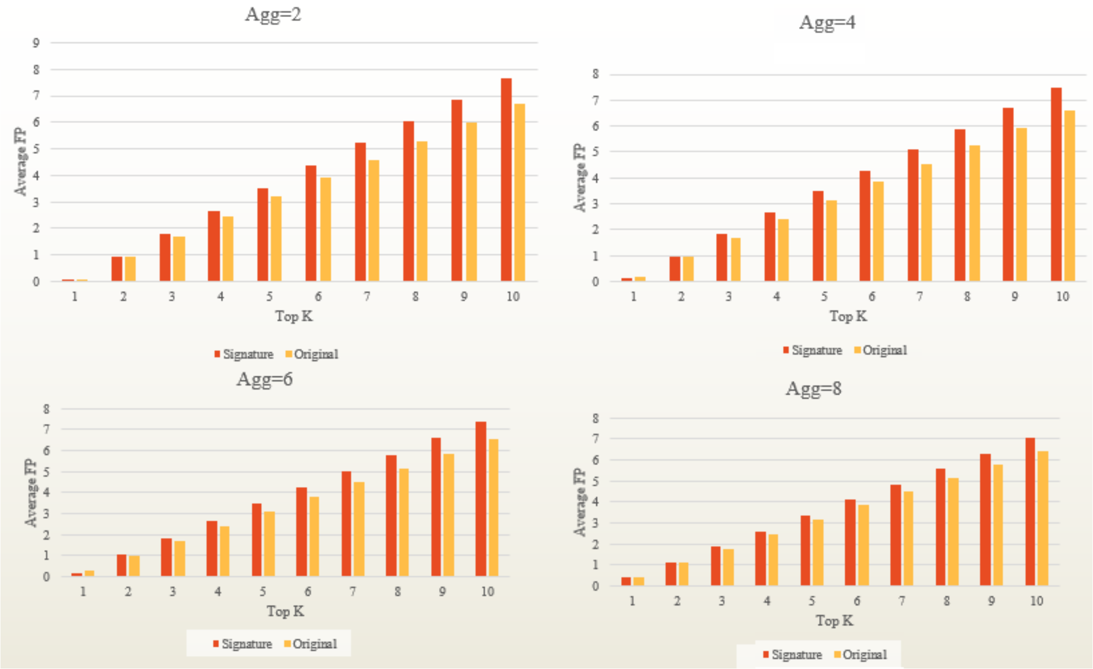


Figure 32. Average of False Positive Values

The average of precision values is shown in figure 33. We can notice that the average of precision using signatures is always greater than the average of precision using originals and as we increase the value of K, the average of precision goes down.

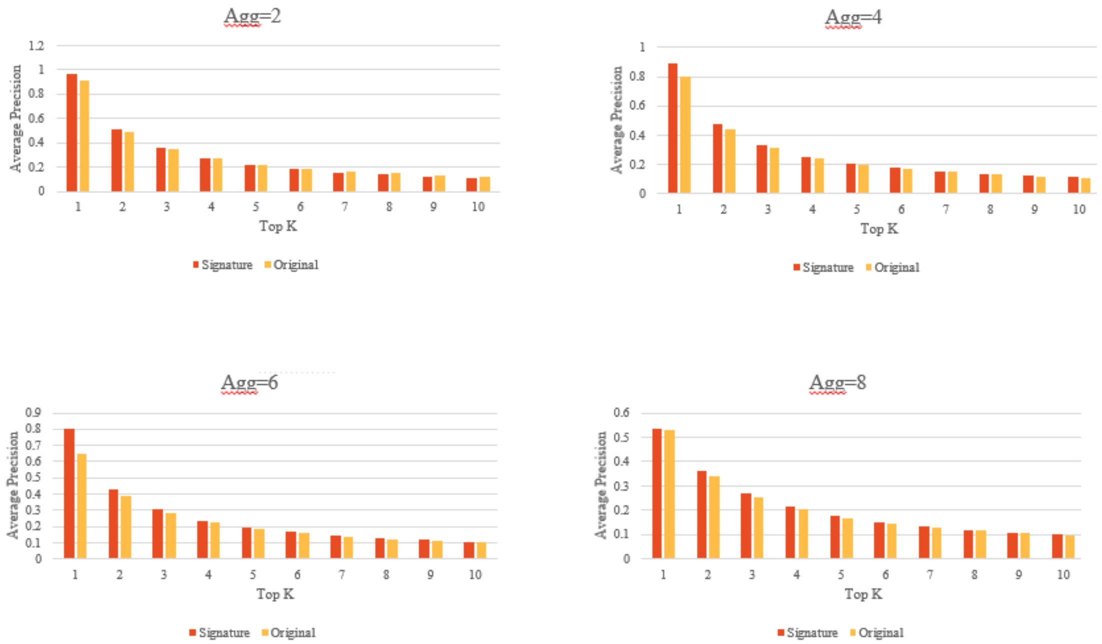


Figure 33. Average of Precision Values

Figure 34 shows the average of true positive values for different aggregation levels for multiple top K values. When we compare an image with its corresponding aggregated image and in order to determine TP, the similarity value of comparing the image with the aggregated image should be within the top k largest values of similarities. As we increase the k values, the TP rate will be increased, and the FP rate will be increased too. To be more accurate, we need to set k to be 1. This means that the similarity of comparing an image with the most corresponding one of the aggregated images should be the largest value of similarity than the value of similarity of comparing an image with all other non-corresponding images. Through the use of signature, the average of TP values is always higher than the average of TP values using original images and aggregated images.

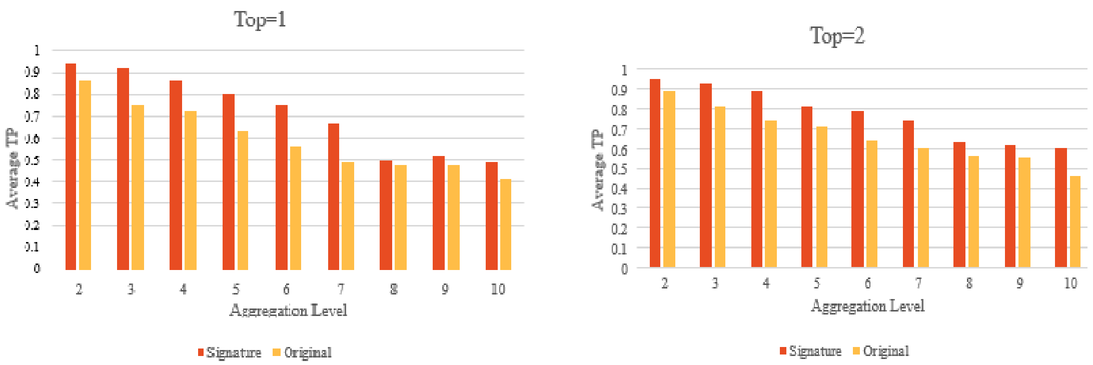


Figure 34. Average of True Positive Values for Different Aggregation Levels

Figure 35 shows the average of FP values for different aggregation levels. It is clear that for most of the aggregation levels and through the use of signature, the average of FP values is always less than the average of FP values using original images and aggregated images.

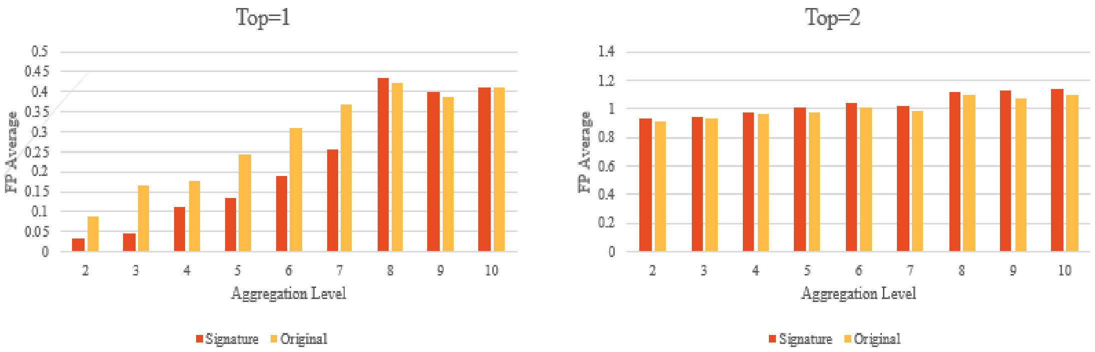


Figure 35. Average of False Positive Values for Different Aggregation Levels

Figure 36 shows the average of precision values for different aggregation levels. For all the aggregation levels and through the use of signature, the average of precision values is always higher than the average of precision values using original images and aggregated images.

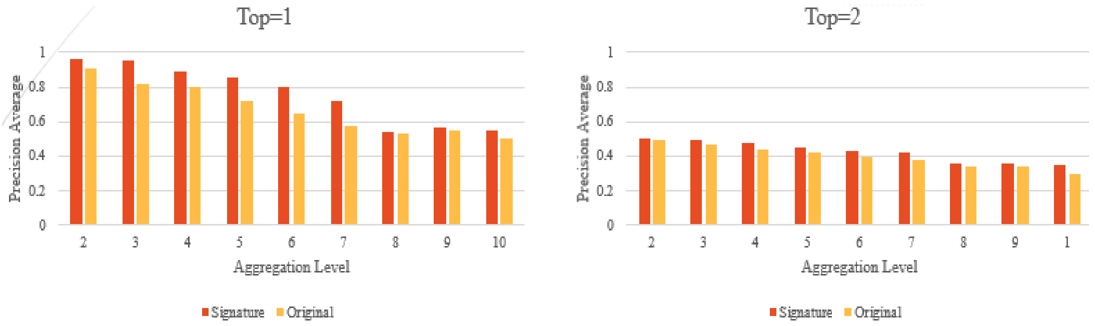


Figure 36. Average of Precision Values for Different Aggregation Levels

Figure 37 shows the area under the curve for TP, FP and precision values for different aggregation levels. For each aggregation level, we took the value of TP, FP and precision for different top k level. For example, to get the area under the curve for TP at aggregation level equals 2, we took the TP values at aggregation level 2 for top k=1, top k=2…top k=10, then we built the percentile curve and then we built the area under the curve. As shown in the figure, the signature space performance is better than the original space. The TP and precision values are better than original.

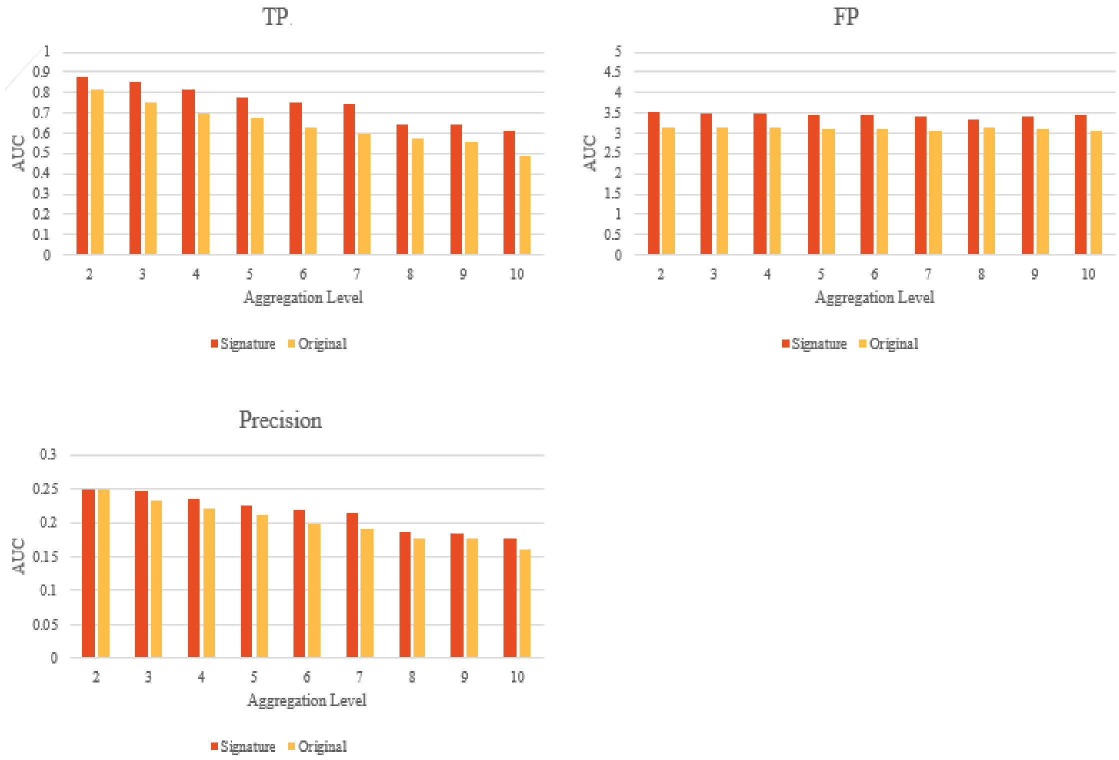


Figure 37. Area Under the Curve for TP, FP, and Precision

Figure 38 shows the area under the curve for TP, FP and precision values for different top k values. For each top k value, we took the value of TP, FP and precision for different aggregation levels. For example, to get the area under the curve for TP at top k=1, we took the TP values at top k=1 for aggregation level=2, aggregation level=3… aggregation level=10, then we built the percentile curve and then we built the area under the curve. As shown in the figure, it is clear that the signature space performance is better than the original space. The TP and precision values are better than original. We provide the area under the curve for different top k values. However, the most important case is when top k=1 and in this case the signature achieves higher top TP rate, lower FP rate and higher precision rate as compared to the original.

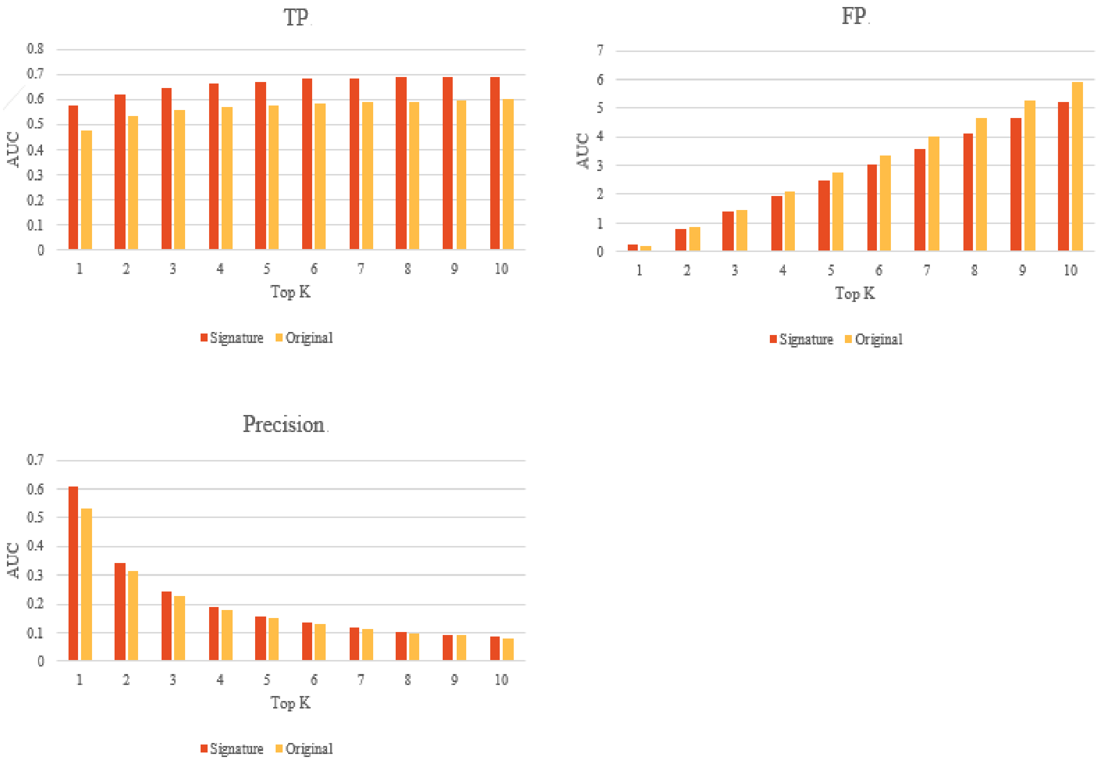


Figure 38. Area Under the Curve for Different Top K Values

3.3.3 Comparing with Related Approaches

After we designed our aggregation sustainable signatures and proved that it outperforms the original data space, then we need to compare it with other related methods including, max-pooling, low pass filtering and Haar wavelet. Max-pooling is utilized in CNN and helps in extracting low level features such as points and edges. It can be used to reduce dimensionality of the dataset. For example, 16 x 16 dataset can be reduced into 8 x 8 dataset through the use of 2 x 2 max-pool. On the other hand, low pass filtering can be considered as the bases of most smoothing methods. It deals with frequencies and preserves the frequencies that are below the cut-off frequency. It also attenuates frequencies that are higher than the cut-off frequency. Haar wavelet is a multi-stage process that deals with frequency and location of the dataset.

In the following experiment, we used a digit dataset that contains 10 classes (0 to 9) and each class contains 250 images. So, the aggregated repository contains 2500 images at each aggregation level. Our task was to index 100 images to the corresponding images from the repository at each aggregation level. The results of TPs, FPs and precisions are shown in figures 39, 40 and 41 respectively. From Figure 39, we see that our signature is good at all aggregation levels. Additionally, our signature is the best at high aggregation levels such as 12, 14, 16, 18 and 20. On the other hand, our signature has low rate of false positives as shown in figure 40, especially at higher levels such as 12, 14, 16, 18 and 20. The same behavior applies to the precision since our signature is good at all aggregation level and it is the best at higher aggregation levels as shown in Figure 41.

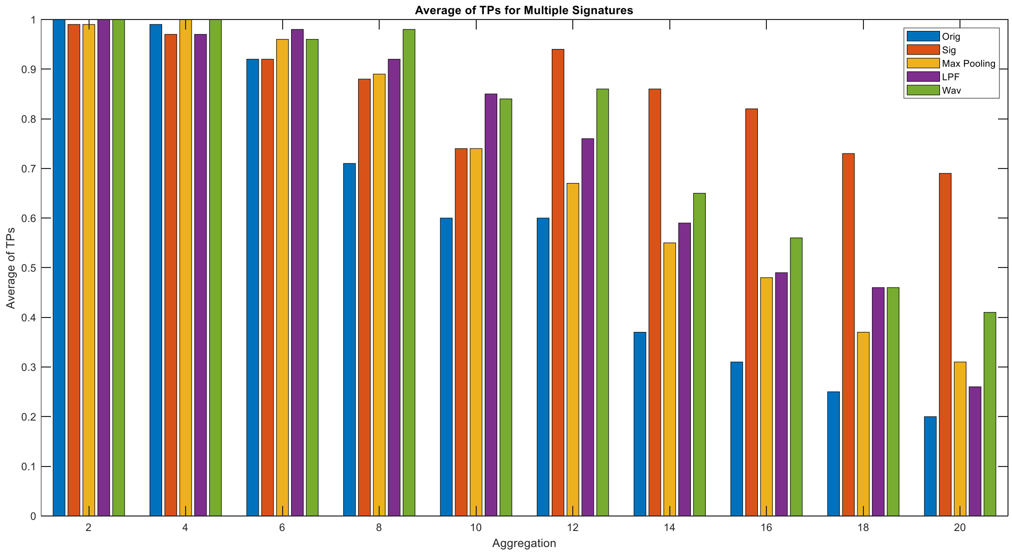


Figure 39. Average of TPs for Multiple Signatures

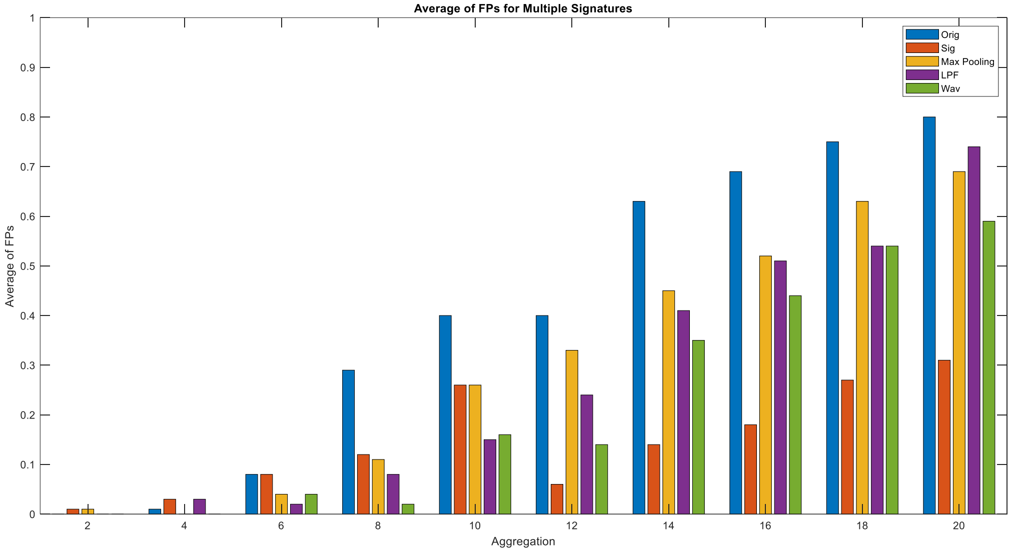


Figure 40. Average of FPs for Multiple Signatures

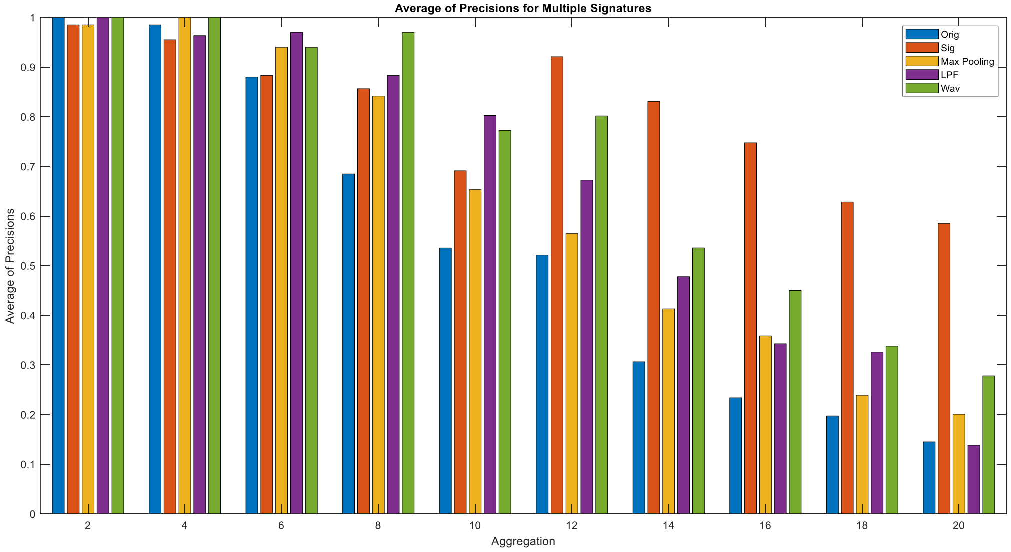


Figure 41. Average of Precisions for Multiple Signatures

Apparently, there is a diversity in the performance of the signatures. There are some signatures are good for fine granularity and our signature is good for high granularity. We address this diversity by developing hybrid signatures that combine our signature with other signatures. Therefore, we take the advantage of our signature at high levels of aggregation and the advantage of other signatures at low levels of aggregation.

3.3.4 Hybrid Signatures

The aim of hybrid signatures is to find a signature that is good on average, which means that it is good at low and high aggregation levels but not the best one as shown in Figure 42. This can be done through the development of hybrid signatures, which combine our signature with other signatures such as max-pooling, low pass filtering and Haar wavelet.

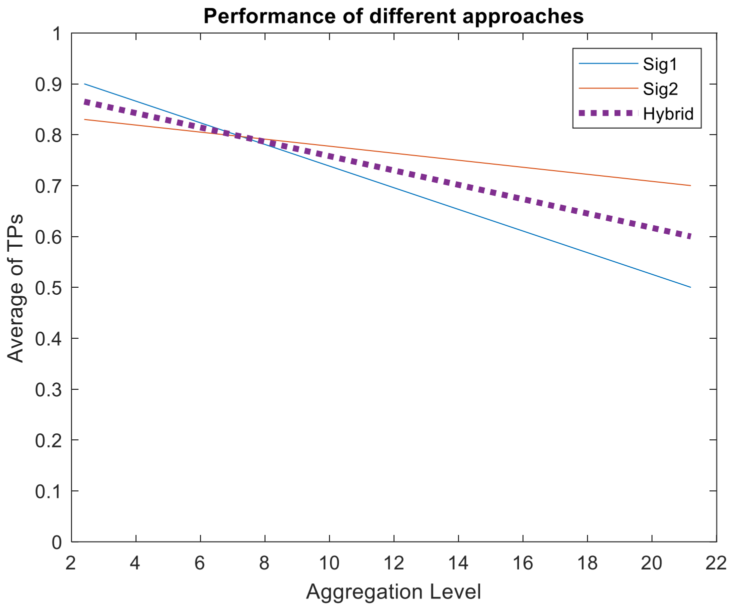


Figure 42. Expected Hybrid Signature Performance

As shown in Figure 43, there are different strategies to combine multiple signatures and to create a hybrid one. For example, we can create the aggregation sustainable data signature

(ASDS) of the original and then find the max-pooling for the resulted signature. another strategy is to find the max-pooling of original and then find the ASDS for the resulted signature. the same thing applies for other methods as shown in Figure 43. Therefore, we have 6 hybrid signatures, including Sig of Max, Max of Sig, Sig of LPF, LPF of Sig, Sig of Wav and Wav of Sig.

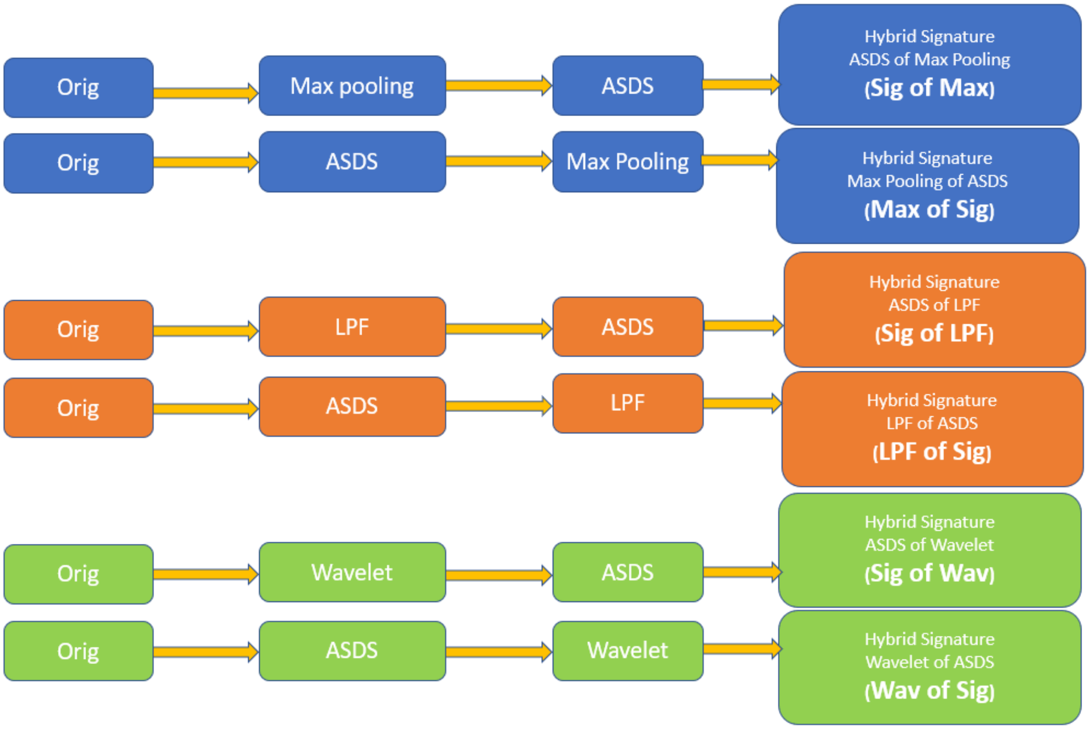


Figure 43. Hybrid Signature Design

In order to select the best strategy for creating the hybrid signature, we need to choose of the two hybrid signatures within each group. For example, we need to choose Sig of Max or Max of Sig. We conducted an experiment for indexing and the results are shown in Figure 44.

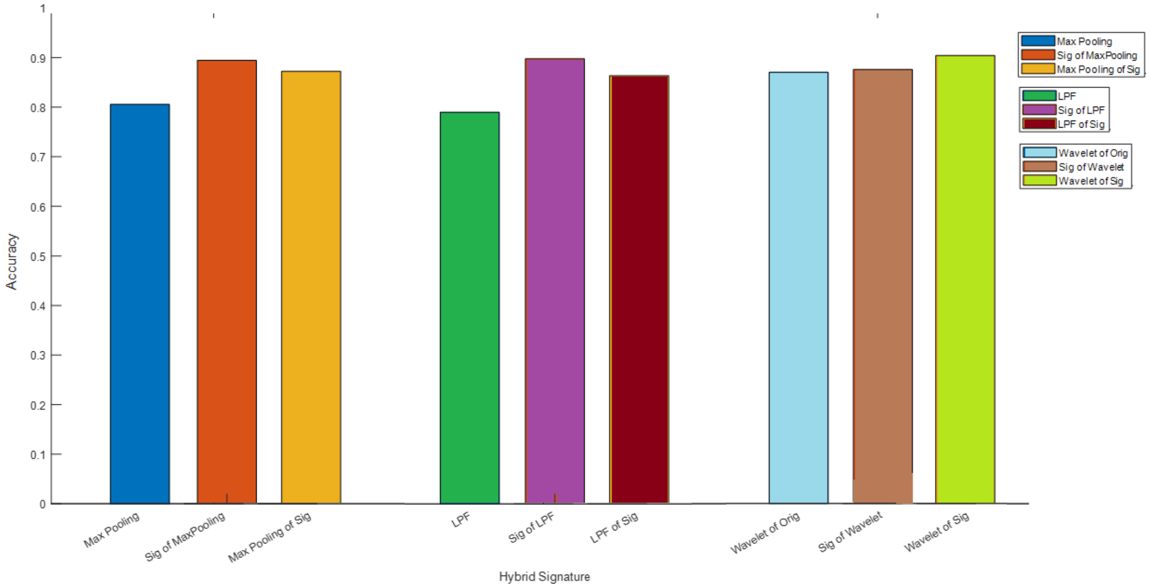


Figure 44. Hybrid Signatures Performance

From Figure 44, we can conclude that the best hybrid signatures are: Sig of Max, Sig of LPF and Wav of Sig. We extended the previous experiment of indexing to include the hybrid signatures and the results of TPs, FPs and Precisions are shown in Figures 45, 46 and 47 respectively. from these figures, we can see that Sig of LPF is good on average and this means that it is good at low and high aggregation levels. Additionally, we can see that Sig of Max can be used at low and high aggregation levels.

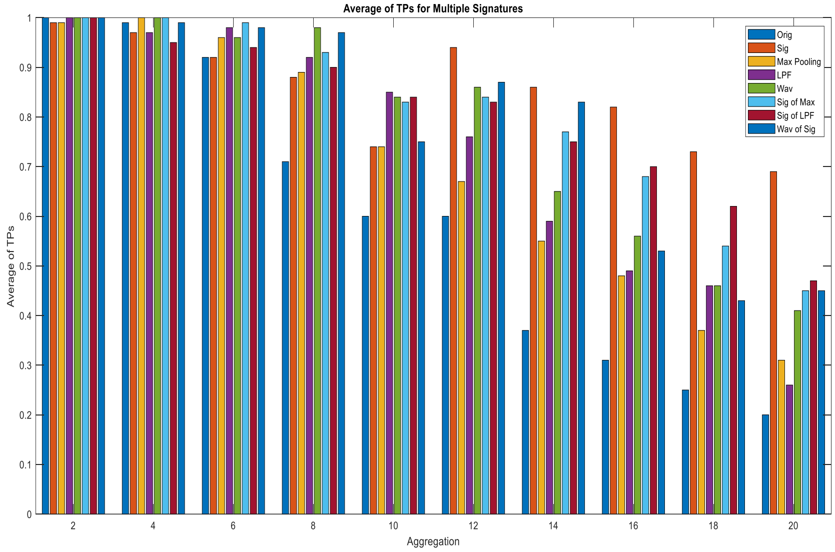


Figure 45. Average of TPs for Multiple Signatures with Hybrid Signatures

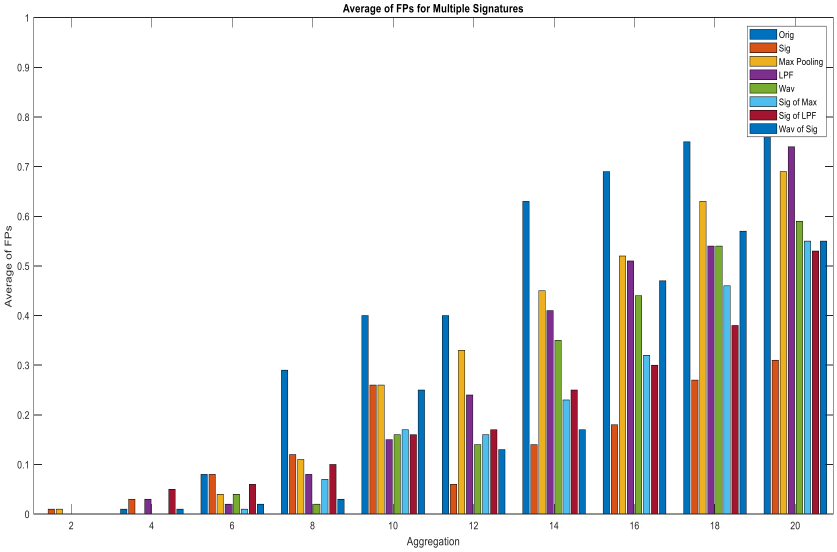


Figure 46. Average of FPs for Multiple Signatures with Hybrid Signatures

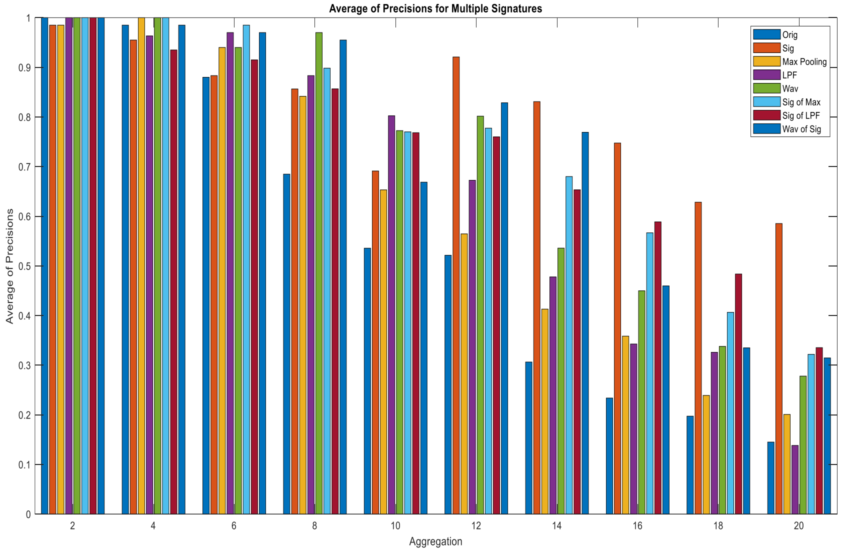


Figure 47. Average of Precisions for Multiple Signatures with Hybrid Signatures

Figure 48 shows the results of exploring TPs, FPs and Precisions in different ranges of aggregation. We started with particular ranges and then we used wider ranges until we reach the final general range, which includes all aggregation levels from low to high as shown in the bottom of Figure 48. The first row in Figure 48 shows that for fine granularities such as 2, 4, 6 and 8, the best signatures are original, max-pooling, low pass filter and wavelet respectively. However, our signature is the best for all high aggregation levels such as 12, 14, 16, 18 and 20, which are the most challenging ones. For this reason, our signature could achieve good performance. The last row in Figure 48 shows TPs rate, FP rates, and precisions for all signatures. We can see that our signature has the highest rate of TP, lowest rate of FP and highest rate of precisions. Our signature's performance is also extremely far away from all other signatures and the heat maps colors prove this performance.

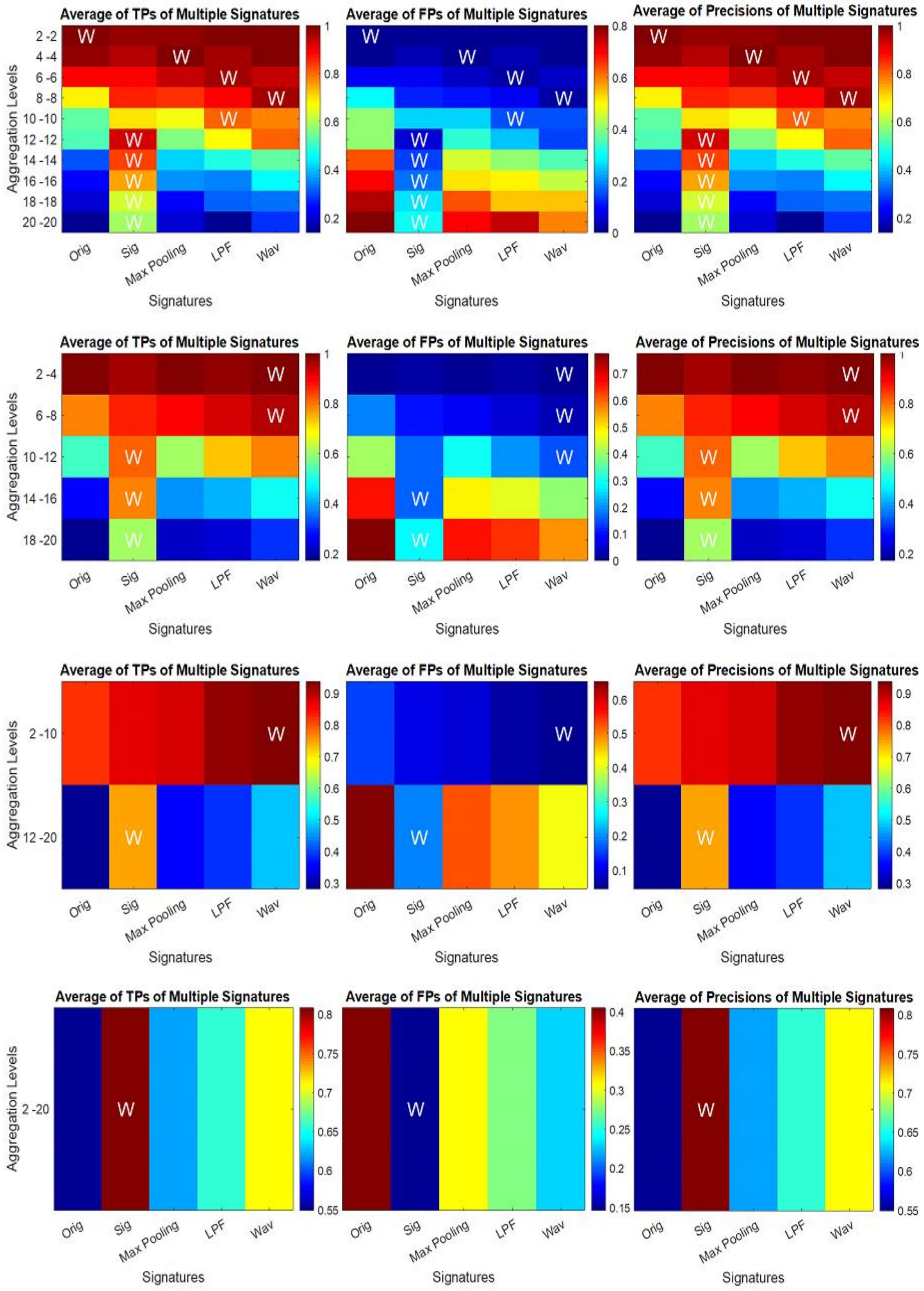


Figure 48. Performance Measures for Multiple Signatures

Figure 49 shows the results of exploring TPs, FPs and Precisions in different ranges of aggregation for all signatures including hybrid signatures. We started with particular ranges and then we used wider ranges until we reach the final general range, which includes all aggregation levels from low to high as shown in the bottom of Figure 49. The first row in Figure 49 shows that for fine granularities such as 2, 4, 6 and 8, the best signatures are original, max-pooling, low pass filter and Sig of Max-pooling respectively. The second row in Figure 49 shows the results for wider ranges and we can notice that Wav of Sig is good for fine granularities. However, our signature is the best for all high aggregation levels such as 12, 14, 16, 18 and 20, which are the most challenging ones. For this reason, our signature could achieve good performance. The last row in Figure 49 shows TPs rate, FPs rate, and precisions rate for all signatures. We can see that our signature has the highest rate of TP, lowest rate of FP and highest rate of precisions. Our signature's performance is also extremely far away from all other signatures and the heat maps colors prove this performance.

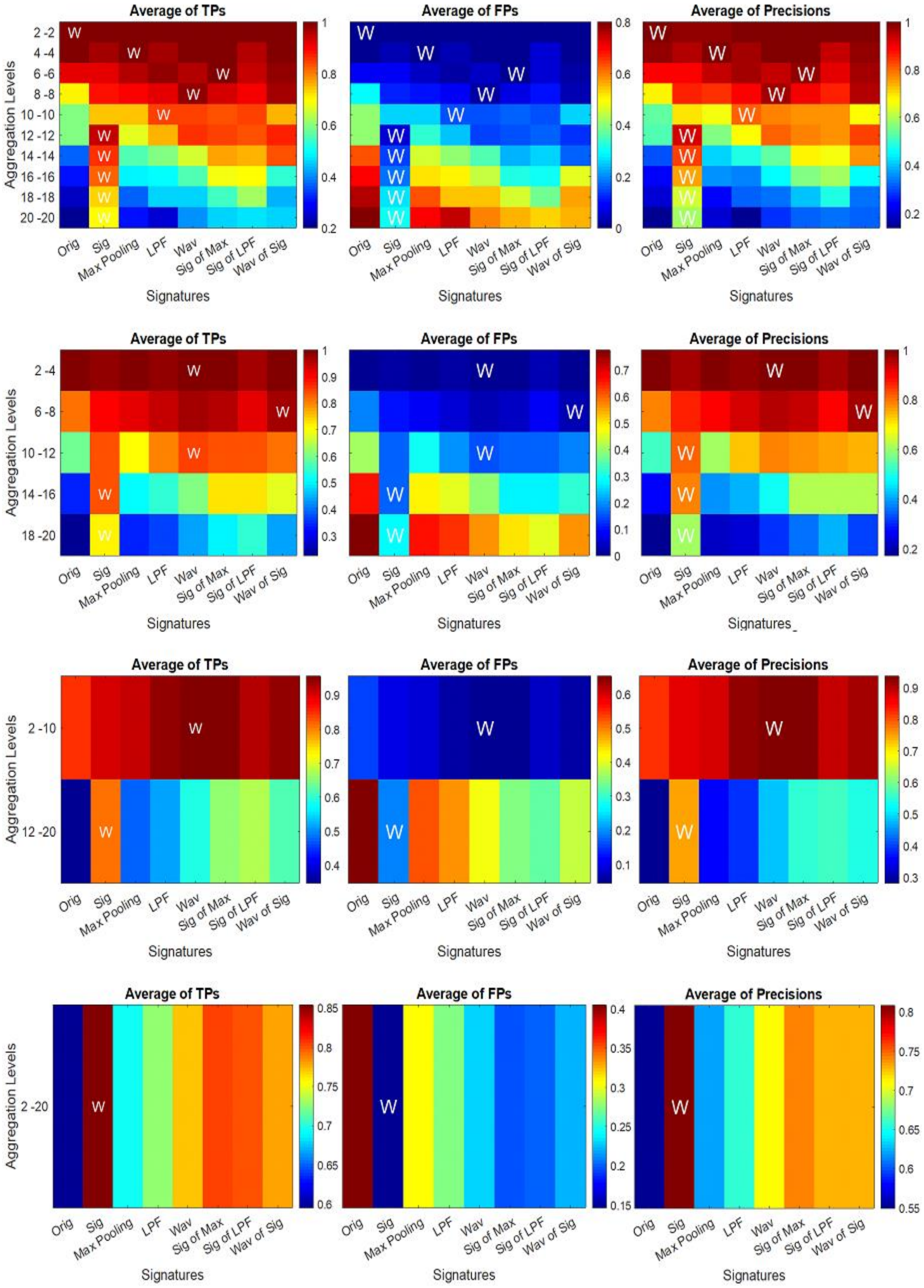


Figure 49. Performance Measures for Multiple Signatures with Hybrid Signatures

Finally, we compared our signature with related approaches, including max-pooling, low pass filtering and haar wavelet and we observed the efficiency of our approach SigMatch in retrieving and indexing aggregated images using different aggregation levels since it could achieve higher accuracy in retrieving related aggregated images. We also created hybrid signatures that combine our signature with other signatures. These hybrid signatures are good on average and this means that they are good in all aggregation levels including low and high aggregation levels.

3.3.5 Practitioner Guide

Given a detailed image, which needs to be indexed to related images in an aggregated repository of images, then we need to choose the best approach for indexing based on the aggregation level(s) of the repository images. The approaches of indexing include original, max-pooling, low pass filtering, wavelet and hybrid signatures (sig of max, sig of LPF, and wav of sig). Our recommendation is to use the following rules in order to select the best indexing method:

- If the aggregation level is very low such as 2 or 4, then it is good to use either original or max-pooling.
- If the aggregation level is low such as 6 or 8, then we recommend to use LPF, Sig of Max and Wav of Sig.
- If the aggregation level is high such as 10 or 12, then we recommend to use our basic signature, LPF or wavelet.
- If the aggregation level is very high such as 14, 16 or more, then we recommend to use our basic signature.

- If the aggregation level is unknown, then we recommend to use our basic signature or the hybrid signatures (Sig of Max and Sig of LPF).

3.4 RESEARCH QUESTION 3: HOW TO BUILD AN EFFICIENT RETRIEVAL ARCHITECTURE FOR AGGREGATED TIME SERIES?

Nowadays, we have a lot of applications that explore time series of different kinds. Those time series are blindfold, large scale and mission critical, such as economical time series, social time series and medical time series. In this research question, we need to extend the aggregation sustainable signature-based approach developed in research question 2 to time series datasets. We explored how we can process and retrieve aggregated time series. Therefore, we considered high frequency time series generated from the dataset in the aggregated repository. Also, if there is a big aggregated timeseries dataset, then how we can relate a certain part of this big aggregated timeseries with a detailed timeseries dataset.

Time series dataset is represented as data points within successive times as shown in Figure 50. Time series data originally emerge from observations that represent evolution of some phenomena over time [43], which allows time series data to provide valuable information about the relationship and dependency between successive observations. Therefore, time series can be widely used to represent trends and fluctuations in different fields such as, finance, weather, astronomical observations and medical observations like blood pressure and body temperature. Time series data are most commonly visualized through line graphs.

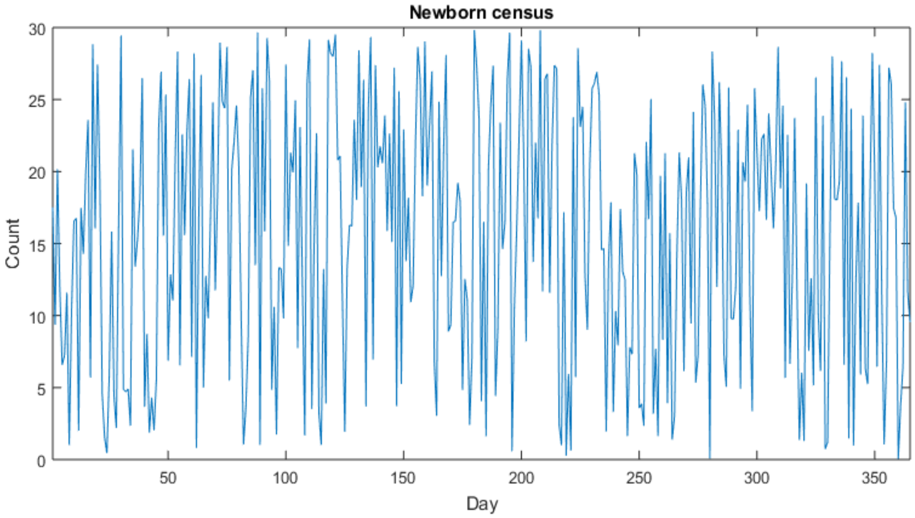


Figure 50. Time Series Example

Figures 51 and 52 show examples of intensive and sparse time series datasets. In Figure 51, we can see the hourly temperatures of a patient within one year and there are 8760 data points. However, the monthly body weight of the same patient is shown in Figure 52 and there are 12 data points. Therefore, by comparing these time series datasets, we can notice the difference between intensive and sparse time series datasets. The intensive time series dataset means high frequency dataset, which implies that the data are collected at a fine scale. Sparse time series dataset means low frequency dataset, which implies that the data are collected at a large scale.

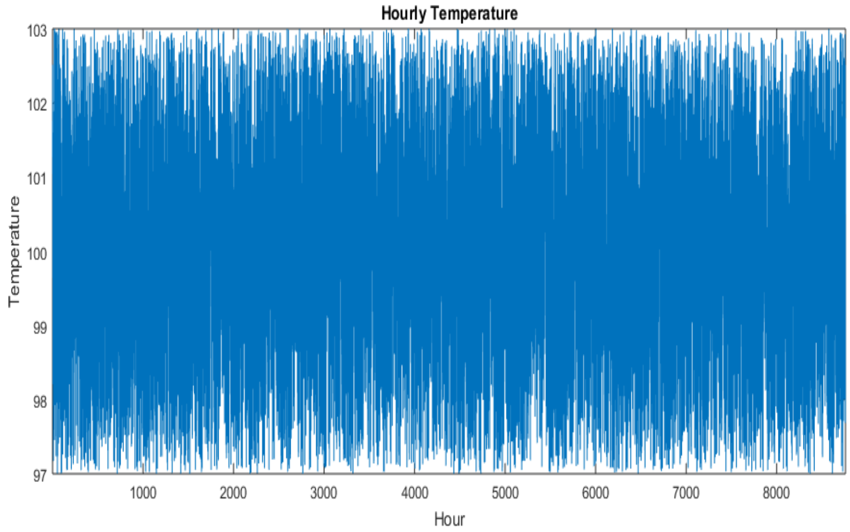


Figure 51. Intensive Time Series Example

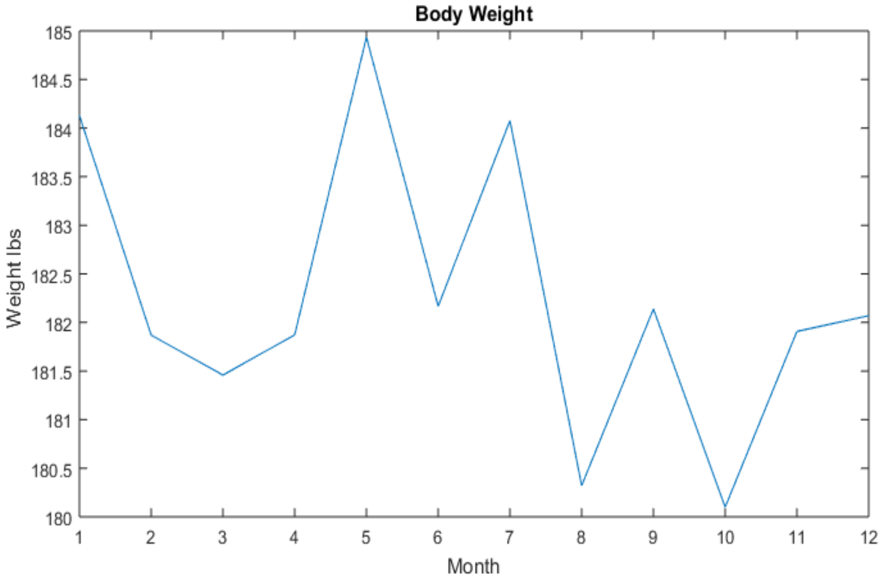


Figure 52. Sparse Time Series Example

There are several measurements to assess the similarity between time series datasets including, Euclidean distance, time series normalization [44], transformation rules [45], dynamic time warping [46], and longest common subsequence [47].

In our work, we used low pass filter [48]. The low pass filter can pass low frequencies and block the high frequencies. Figure 53 shows an example of low pass filter using different cut-off frequencies. From Figure 53, we can notice that decreasing the cut-off frequency will result in blocking more higher frequencies. On the other hand, increasing the cut-off frequency will result in passing more frequencies since these frequencies will be less than the cut-off frequency.

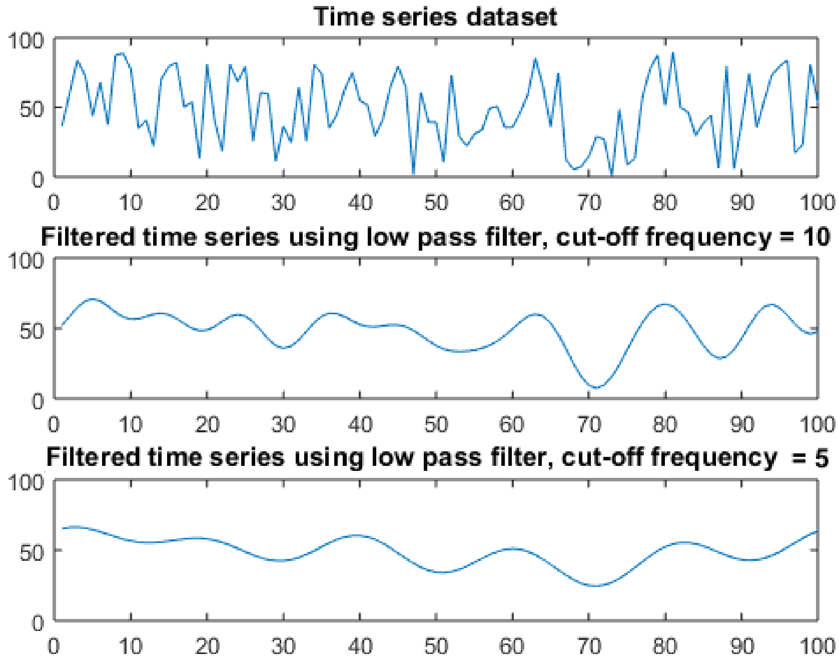


Figure 53. Example of Low Pass Filter

As shown in Figure 54, we assessed the similarity between time series datasets using four different paths including:

1. Compare the two time series datasets (T1 and T2) using a distance measure (path P1).

2. Compare the two signatures (Sig1 and Sig2) of the two time series datasets using a distance measure (path P2).
3. Compare the two transformed time series datasets (TSF1 and TSF2) using a distance measure (path P3).
4. Compare the two hybrid signatures of the transformed time series datasets (TSF1_Sig and TSF2_sig) using a distance measure (path P4).

The distance measure can be different based on the properties of the time series dataset. For example, the transformed time series dataset will be different than the original time series dataset, and therefore the distance measure will be different.

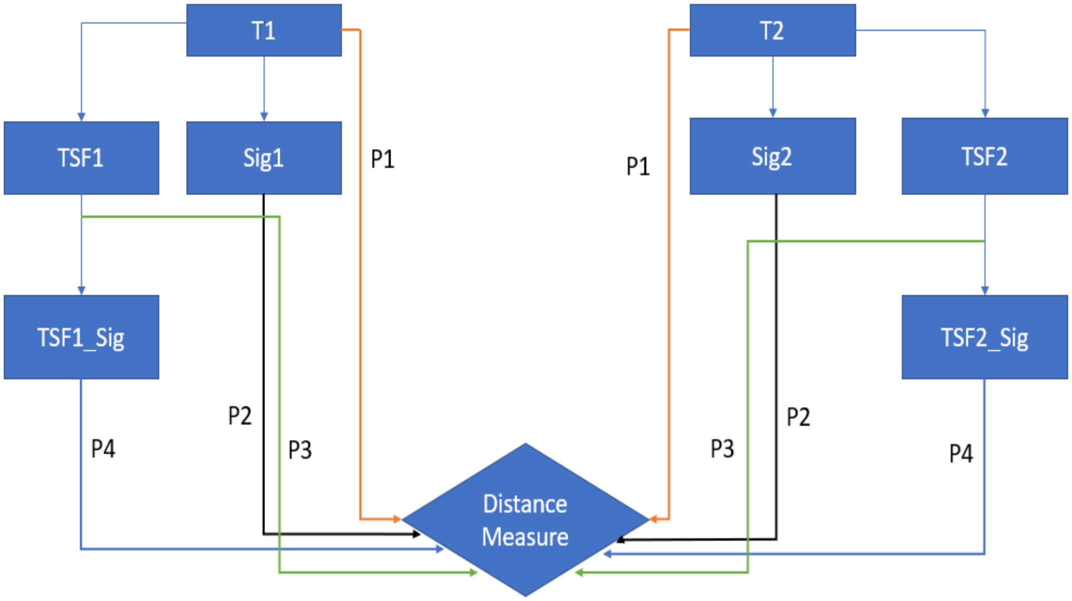


Figure 54. Time Series Comparison Approaches

In our work, when we have two different (in size) time series datasets, we use the dynamic time wrapping measure as a measurement to assess the similarity between time series datasets. As an example of time series dataset to be compared is the monthly stock sales with daily stock sales. In this case, we have a raw dataset and aggregated time series dataset. Another example to compare is the daily temperature dataset with historical datasets from previous years.

1. If we have big datasets, we use a sliding window technique and treat each window as a sperate time series dataset and follow the paths as shown in Figure 42.
2. Rank the results in a descending order.
3. Get top K results.

We compared our proposed approach with advanced time series processing techniques such as max-pooling, low pass filtering and wavelets decomposition [49]. Finally, we developed a strategy to process and utilize aggregated data using aggregation sustainable signatures.

3.4.1 Signature Design

We extended the aggregation sustainable signature that we created in research question 2 to time series datasets. As shown in Figure 55, the filter is one dimension, we changed the filter to be divided into 3 parts instead of 2 parts, including left part, central part and right part. the left and right parts have the same size. The size of the central part can be in the form of (2x + 1), where x > 1.

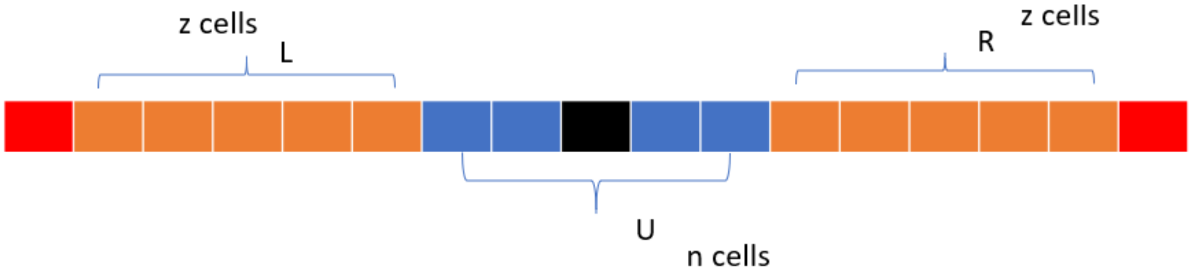


Figure 55. Signature Design for Time Series

The signature can be created using the following equations:

$$Um = \frac{1}{n} \sum_{q=i-(n-1)/2}^{i+(n-1)/2} m(i) e^{-|i-q|}$$

$$Lm = \frac{1}{z} \sum_{q=i-(n-1)/2-z}^{i-(n-1)/2} m(i) e^{-|i-q|}$$

$$Rm = \frac{1}{z} \sum_{q=i+(n-1)/2}^{i+(n-1)/2+z} m(i) e^{-|i-q|}$$

$$\theta = \tan^{-1}((Rm - Lm) / n)$$

$$Signature(i) = (1 + Um)^{v} (1 + \sin(\theta)(|Rm - Lm|) + \cos(\theta)(Rm + Lm))^{p}$$

Where n is the size of the central part, z is the size of the left and right parts.

In order to create a signature for a time series dataset, this scanning filter needs to be applied on each cell of the dataset. The black cell in the center of the filter will be applied to the time series data cell (m(i)).

3.4.2 Experimental Study

One example of time series dataset is EEG dataset, in which Each column (attribute) represents a time series dataset. An EEG, or Electroencephalogram, is a test that records the electrical signals of the brain using small metal discs (electrodes) that are attached to the scalp [44]. The brain cells communicate with each other using electrical impulses, which are always working, even if the person is asleep. The brain activity will show up on an EEG reading as wavy lines, which is a snapshot in time of the electrical activity in your brain as shown in Figure 56.

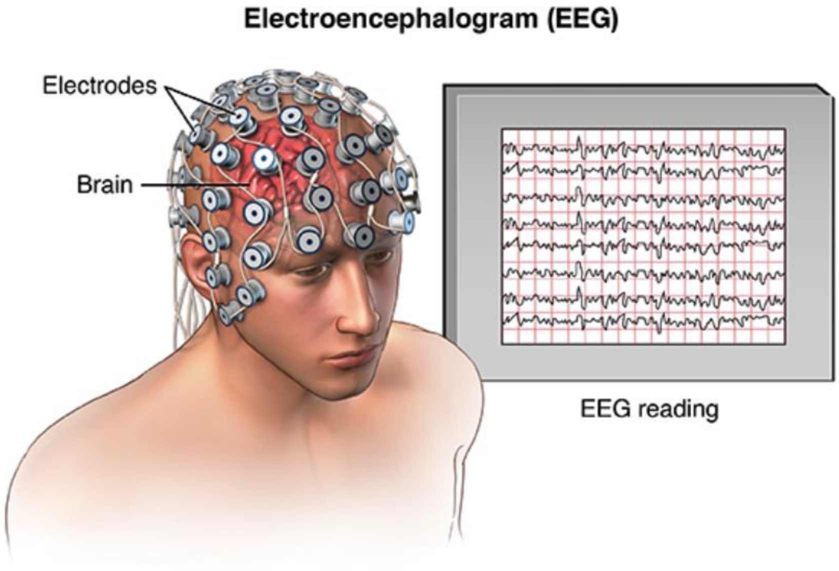
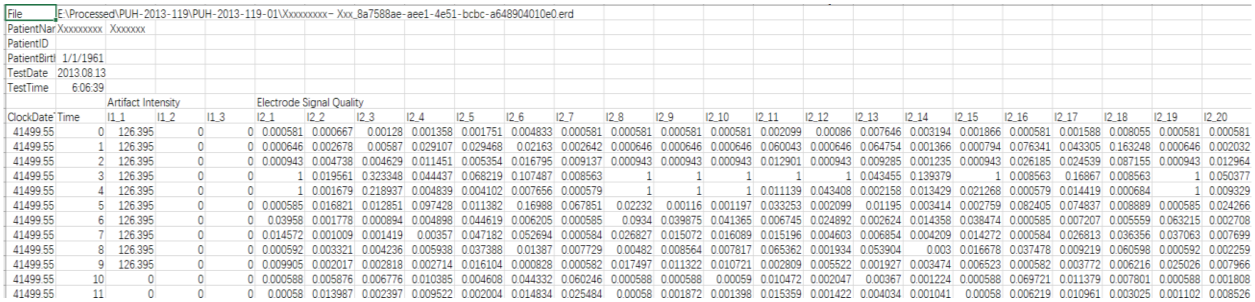


Figure 56. EEG Brain Data

http://sipl.eelabs.technion.ac.il/projects/estimating-brain-activation-patterns-from-eeg-data/

Wavy lines will be written in CSV format. Figure 57 provides an example of EEG dataset. Each patient has around 6000 attributes, which are recorded every second. The size of CSV files is up to 4 GB and every patient typically has ~10 files with an overall size of about ~10 TB.



File	E:\Processed\PUH-2013-119\PUH-2013-119-01\Xxxxxxxxx– Xxx_8a7588ae-aee1-4e51-bcbc-a648904010e0.erd																							
PatientNar	Xxxxxxxxx	Xxxxxxx																						
PatientID																								
PatientBirtl	1/1/1961																							
TestDate	2013.08.13																							
TestTime	6:06:39																							
		Artifact Intensity			Electrode Signal Quality																			
ClockDate	Time	I1_1	I1_2	I1_3	I2_1	I2_2	I2_3	I2_4	I2_5	I2_6	I2_7	I2_8	I2_9	I2_10	I2_11	I2_12	I2_13	I2_14	I2_15	I2_16	I2_17	I2_18	I2_19	I2_20
41499.55	0	126.395	0	0	0.000581	0.000667	0.00128	0.001358	0.001751	0.004833	0.000581	0.000581	0.000581	0.000581	0.002099	0.00086	0.007646	0.003194	0.001866	0.000581	0.001588	0.008055	0.000581	0.000581
41499.55	1	126.395	0	0	0.000646	0.002678	0.00587	0.029107	0.029468	0.02163	0.002642	0.000646	0.000646	0.000646	0.060043	0.000646	0.064754	0.001366	0.000794	0.076341	0.043305	0.163248	0.000646	0.002032
41499.55	2	126.395	0	0	0.000943	0.004738	0.004629	0.011451	0.005354	0.016795	0.009137	0.000943	0.000943	0.000943	0.012901	0.000943	0.009285	0.001235	0.000943	0.026185	0.024539	0.087155	0.000943	0.012964
41499.55	3	126.395	0	0	1	0.019561	0.323348	0.044437	0.068219	0.107487	0.008563	1	1	1	1	1	0.043455	0.139379	1	0.008563	0.16867	0.008563	1	0.050377
41499.55	4	126.395	0	0	1	0.001679	0.218937	0.004839	0.004102	0.007656	0.000579	1	1	1	0.011139	0.043408	0.002158	0.013429	0.021268	0.000579	0.014419	0.000684	1	0.009329
41499.55	5	126.395	0	0	0.000585	0.016821	0.012851	0.097428	0.011382	0.16988	0.067851	0.02232	0.00116	0.001197	0.033253	0.002099	0.01195	0.003414	0.002759	0.082405	0.074837	0.008889	0.000585	0.024266
41499.55	6	126.395	0	0	0.03958	0.001778	0.000894	0.004898	0.044619	0.006205	0.000585	0.0934	0.039875	0.041365	0.006745	0.024892	0.002624	0.014358	0.038474	0.000585	0.007207	0.005559	0.063215	0.002708
41499.55	7	126.395	0	0	0.014572	0.001009	0.001419	0.00357	0.047182	0.052694	0.000584	0.026827	0.015072	0.016089	0.015196	0.004603	0.006854	0.004209	0.014272	0.000584	0.026813	0.036356	0.037063	0.007699
41499.55	8	126.395	0	0	0.000592	0.003321	0.004236	0.005938	0.037388	0.01387	0.007729	0.00482	0.008564	0.007817	0.065362	0.001934	0.053904	0.003	0.016678	0.037478	0.009219	0.060598	0.000592	0.002259
41499.55	9	126.395	0	0	0.009905	0.002017	0.002818	0.002714	0.016104	0.000828	0.000582	0.017497	0.011322	0.010721	0.002809	0.005522	0.001927	0.003474	0.006523	0.000582	0.003772	0.006216	0.025026	0.007966
41499.55	10	0	0	0	0.000588	0.005876	0.006776	0.010385	0.004608	0.044332	0.060246	0.000588	0.000588	0.00059	0.010472	0.002047	0.00367	0.001224	0.000588	0.069721	0.011379	0.007801	0.000588	0.001808
41499.55	11	0	0	0	0.00058	0.013987	0.002397	0.009522	0.002004	0.014834	0.025484	0.00058	0.001872	0.001398	0.015359	0.001422	0.004034	0.001041	0.00058	0.006219	0.010961	0.003025	0.001102	0.008526

Figure 57. Example of EEG Dataset

The EEG data can be at very high rate and this means that there is a huge amount of data by the time. In our experiment, there are 26 channels. If the reading is at a rate of 60 Hz, then the amount of data per day equals 26 x 60 x 60 x 60 x 24 readings, which represents a large-scale data that is very difficult to handle in short time. Every channel can measure more than one attribute at the same time and thus the total number of readings will be highly increased. In order to make sense of these data such as comparing these data with historical data to diagnose a patient health status and save his/her life, then we need to process these data in an expediated process, which is completely difficult using a huge amount of data. Therefore, the optimal solution is to use smaller version of the data that can be achieved using aggregation. However, aggregation can result in loss of critical details, which in turn can affect the efficiency of diagnosis and thus threaten the patient's life. Our approach can solve this problem by creating signatures of aggregated data that are very close to the original data and this means that we are using a smaller version of the data that is more informative than the aggregated version.

In our experiment, we have a repository of 1039 patients' datasets. Every dataset has different version according to the aggregation level. For example, version of aggregation level 5 means that we are aggregating the readings of each 5 seconds. Our task is to match 100 raw datasets with their related datasets in the repository at each aggregation level. We used different approaches to perform this task including, original data, max-pooling, low pass filtering and wavelet.

The results of TPs, FPs and precisions are shown in Figures 58, 59 and 60 respectively. From Figure 58, we see that our basic signature is good at all aggregation levels. Additionally, our basic signature is the best at high aggregation levels such as 20 to 48. On the other hand, our signature has low rate of false positives as shown in Figure 59, especially at higher levels such as

20 to 48. The same behavior applies to the precision since our signature is good at all aggregation level and it is the best at higher aggregation levels as shown in Figure 60.

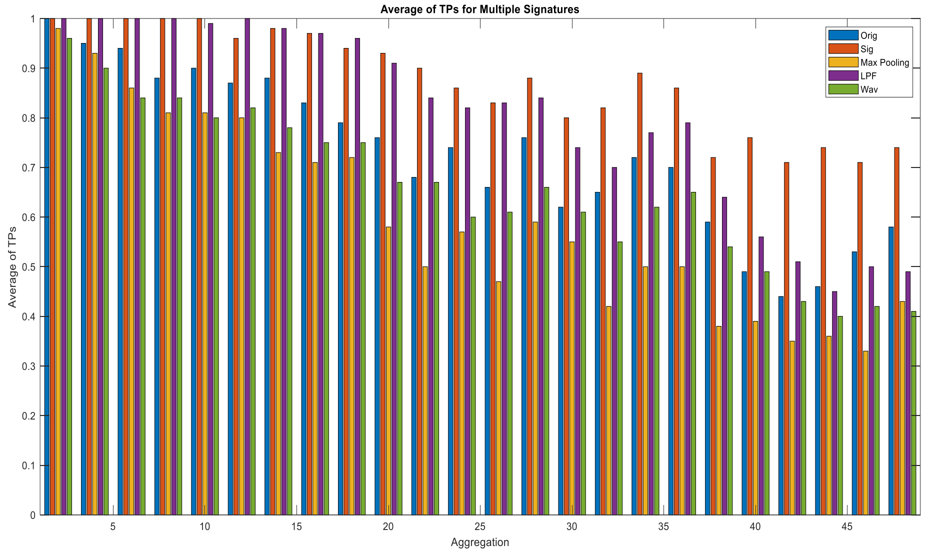


Figure 58. Average of TPs for Multiple Signatures

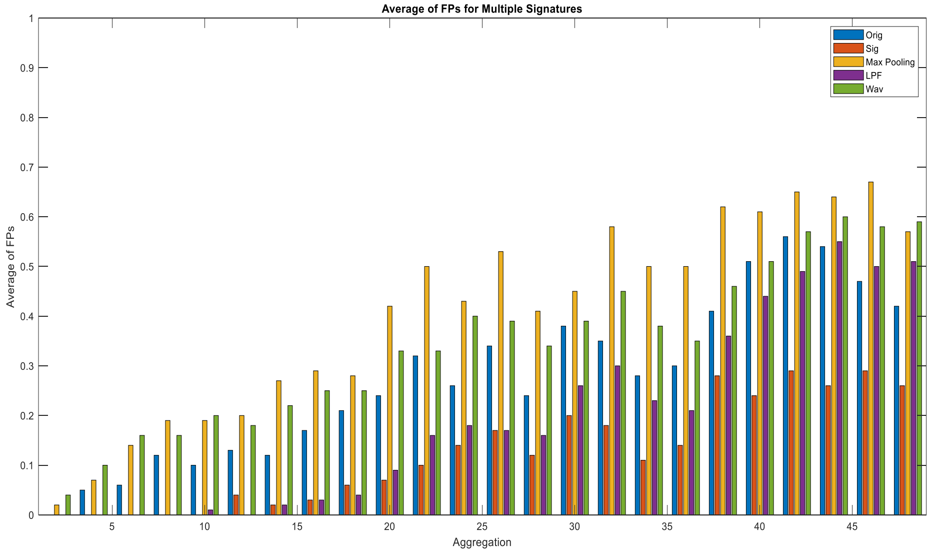


Figure 59. Average of FPs for Multiple Signatures

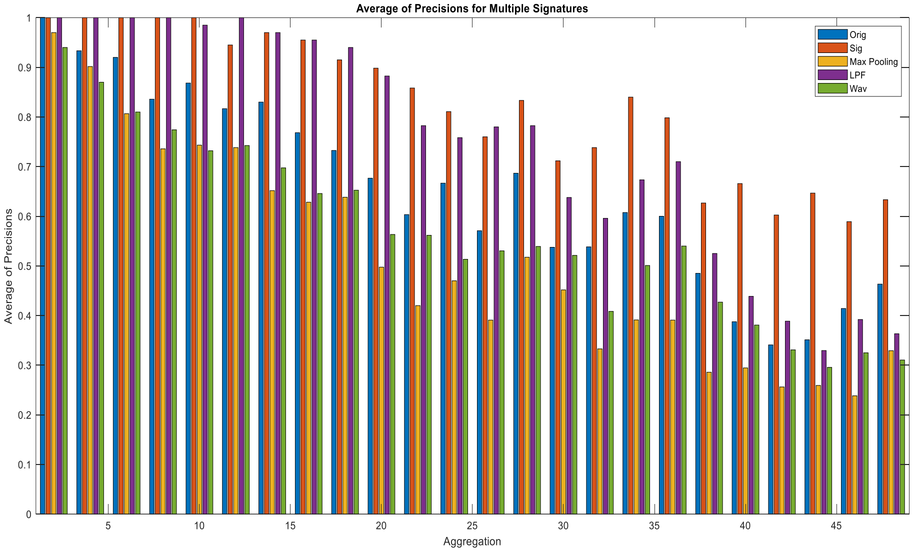


Figure 60. Average of Precisions for Multiple Signatures

From previous figures, we can see that our signature is good at all aggregation levels. The best signatures at the lower aggregation levels are low pass filtering and our signature. For high aggregation levels such as aggregation level 20 to 48, the best signature is our signature since it has the highest TP rate, lowest FP rate and the highest precision rate.

We created hybrid signatures as we mentioned earlier in question 2. The results of TPs, FPs and precisions are shown in Figures 61, 62 and 63. From these figures, we can see that the best signatures are our signature and the hybrid signature Sig of LPF for all aggregation levels including low and high aggregation levels. The Sig of LPF consists of our signature and low pass filtering. It is apparently noticeable that our hybrid signature improved the performance of the indexing process.

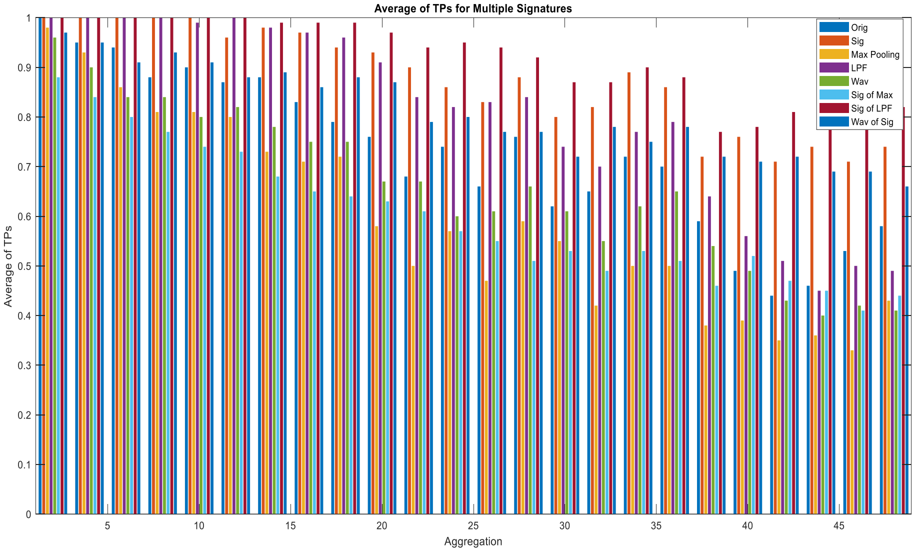


Figure 61. Average of TPs for Multiple Signatures with Hybrid Signatures

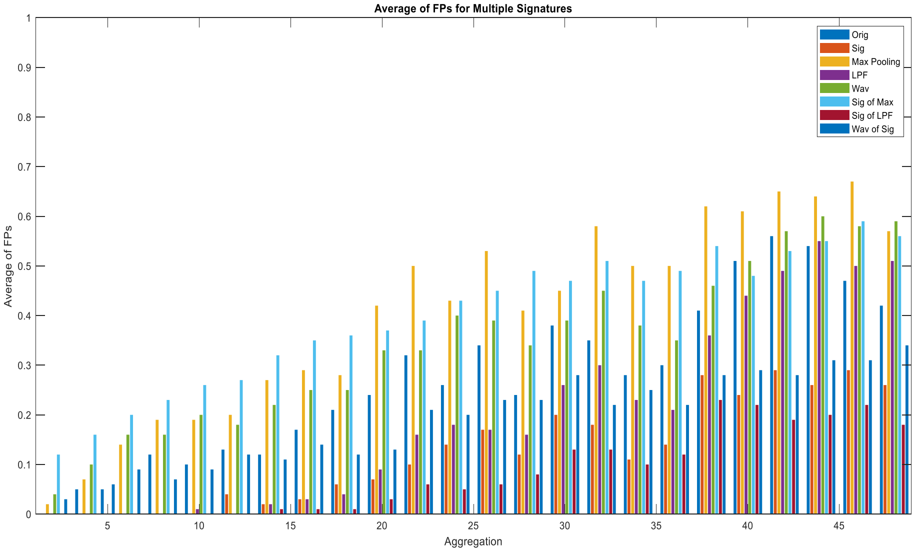


Figure 62. Average of FPs for Multiple Signatures with Hybrid Signatures

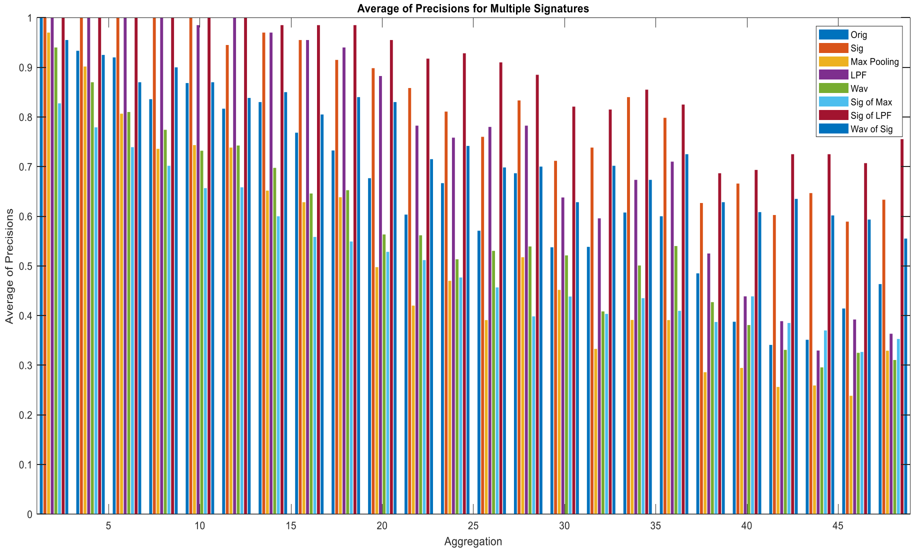


Figure 63. Average of Precisions for Multiple Signatures with Hybrid Signatures

Figure 64 shows the results of exploring TPs, FPs and precisions in different ranges of aggregation. We started with particular ranges and then we used wider ranges until we reach the final general range, which includes all aggregation levels from low to high (2:48) as shown in the bottom of Figure 64. The figure shows that our signature has the best performance at all aggregation levels including low and high ranges.

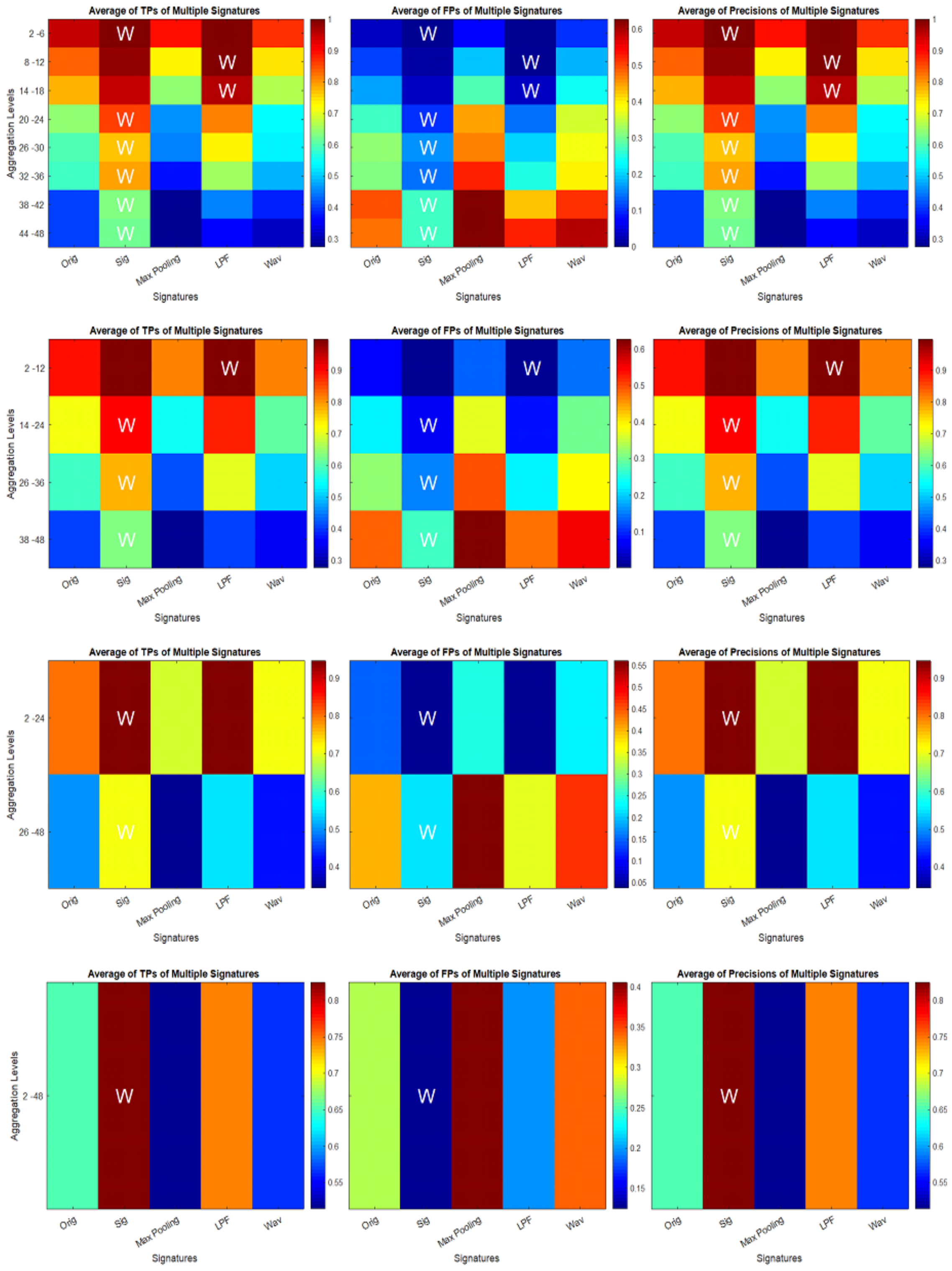


Figure 64. Performance Measures for Different Signatures

Figure 65 shows the results of exploring TPs, FPs and precisions in different ranges of aggregation. We started with particular ranges and then we used wider ranges until we reach the final general range, which includes all aggregation levels from low to high (2:48) as shown in the bottom of Figure 65. The first row in Figure 64 shows that for fine granularities such as 2, 4, 6 and 8, the best signatures are our basic signature and the hybrid signature Sig of LPF. However, the hybrid signature Sig of LPF is the best for all high and low aggregation levels. Our hybrid signature could achieve a good performance as it can be used at any aggregation level. The last row in Figure 65 shows TPs rate, FPs rate, and precisions rate for all signatures. We can see that our hybrid signature has the highest rate of TP, the lowest rate of FP and the highest rate of precision. Our hybrid signature's performance is also extremely far away from all other signatures and the heat maps colors prove this performance. By comparing the heat maps colors, our hybrid Wav of Sig could improve the performance of wavelet signature as shown in Figure 65. As we can see in the last row in Figure 65, Wav of Sig has a yellow color as compared with the blue color of the Wav signature.

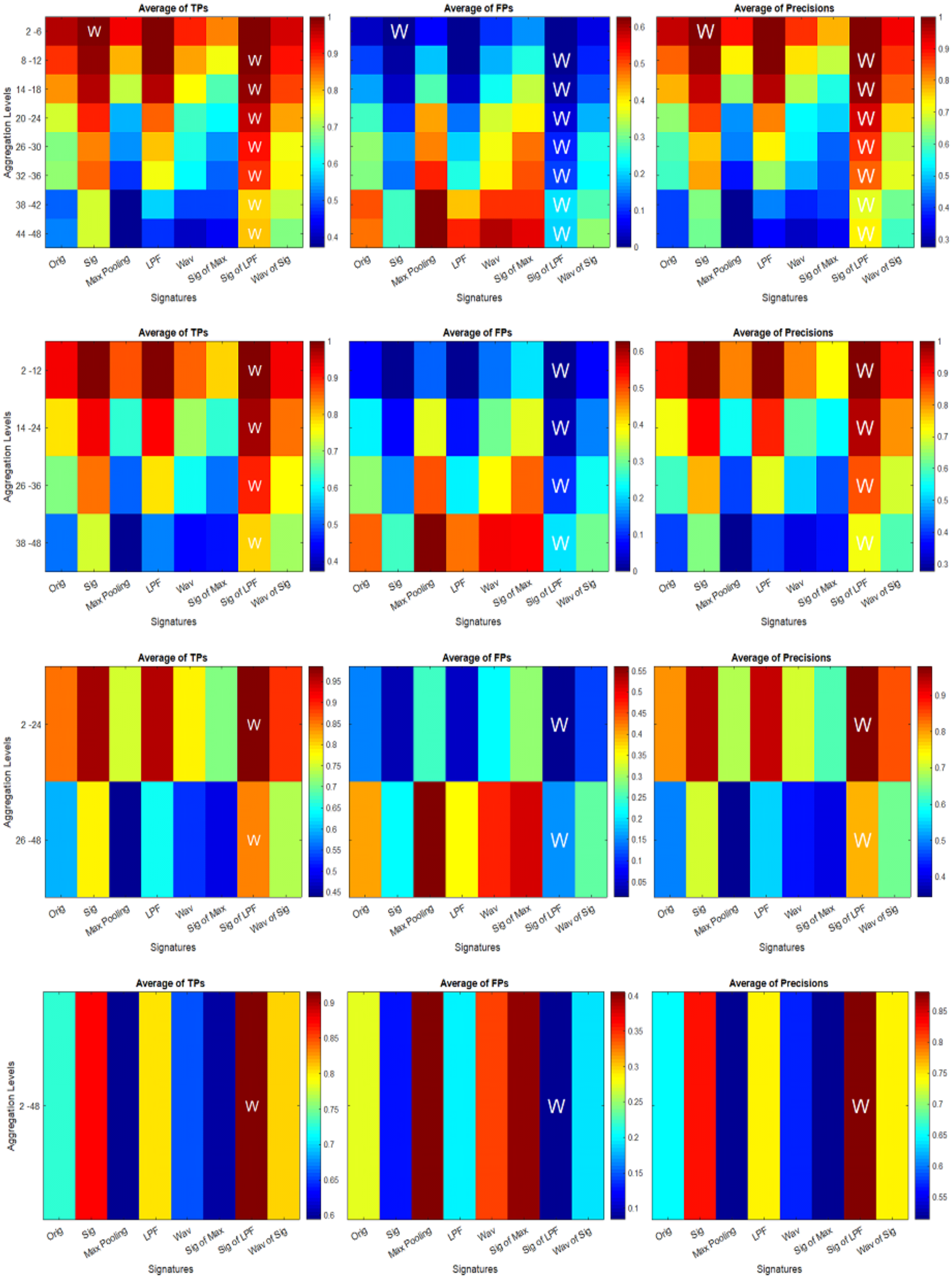


Figure 65. Performance Measures for Multiple Signatures with Hybrid Signatures

In summary, we extended our aggregation sustainable signature to time series datasets and we could achieve significant performance using our basic signature and the hybrid signature that combines our aggregation sustainable signature with low pass filtering. Our experimental results prove that our aggregation sustainable signature is the best when it is used alone. It is also the best when it is combined with low pass filtering and wavelet to create the hybrid signature.

3.4.3 Practitioner Guide

Given a detailed time series dataset, which needs to be indexed to related time series datasets in an aggregated repository, then we need to choose the best approach for indexing based on the aggregation level(s) of the repository time series datasets. In this case, our recommendation is to use Sig of LPF and our basic signature. Our basic signature is the best for fine granularity such as 2, 4 or 6. Sig of LPF outperforms all signatures at all aggregation levels.

4.0 CONCLUSION

We developed an aggregation sustainable signature to improve the quality of data and preserve more information in an aggregated dataset. we used this aggregation sustainable signature to build an efficient aggregated information retrieval architecture using signatures (SigMatch) for images datasets, which could optimize the matching process. We compared our approach with related approaches including, max-pooling, low pass filtering and wavelet and we found that our approach outperforms the other approaches. Based on our analysis, we found that some signatures perform better at certain aggregation levels. In order to improve the overall performance, we developed hybrid approaches to get the advantages from all signatures. We then extended our approach to be used with time series datasets in order to create a representative signature for aggregated time series data. We created heat maps, to be used as practitioner guides to select the best signature(s) according to the aggregation level, for both images and time series datasets. The experimental studies showed the efficiency of our signature and the hybrid signatures. Our approach can be widely applied in the industry where the communication is a vital part since it allows to send smaller version of the data instead of huge amount of data over the network. Additionally, it can filter noisy data.

Bibliography

1. Boubiche, S., et al., *Big Data Challenges and Data Aggregation Strategies in Wireless Sensor Networks.* IEEE Access, 2018. **6**.
2. Han, P., Z. Li, and J. Gong, *Effects of Aggregation Methods on Image Classification.* Geospatial Technology for Earth Observation, 2009: p. 271-288.
3. Hay, G., K. Niemann, and D. Niemann, *Spatial thresholds, image objects, and up-scaling: a multi-scale evaluation.* Remote Sensing of Environment, 1997. **62**: p. 1-19.
4. Collins, J. and C. Woodcock, *Geostatistical estimation of resolution-dependent variance in remotely sensed image.* Photogrammetric Engineering and Remote Sensing, 1999. **65**: p. 41-51.
5. Wang, G., G. Gertner, and A. Anderson, *Up-scaling methods based on variability weighting and simulation for inferring spatial information across scales.* International Journal of Remote Sensing, 2004. **25**(22): p. 4961-4979.
6. Bengio, Y., *Learning deep architectures for AI.* Foundations and trends in Machine Learning, 2009. **2**(1).
7. yıldırım, O. and U. Baloglu, *REGP: A New Pooling Algorithm for Deep Convolutional Neural Networks.* Neural Network World 2019. **29**(1).
8. Li , T., et al., *Robust geometric ℓp-norm feature pooling for image classification and action recognition.* Image and Vision Computing, 2016. **55**(2).
9. Mcnames, J. and B. Goldstein, *A nonlinear lowpass filter that eliminates peak attenuation*, in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference*. 2002, IEEE.
10. Alshareefi, N., *Image compression using wavelet transform.* Journal of Babylon University/Pure and Applied Sciences, 2013. **21**(4).
11. *The Importance of Disaggregated Data. Child & Youth Health*. 2009, National Collaborating Centre for Aboriginal Health.
12. Nakano, Y., *Non-Intrusive Electric Appliances Load Monitoring System Using Harmonic Pattern Recognition*. 2004.
13. Patel, S., *At the Flick of a Switch: Detecting and Classifying Unique Electrical Events on the Residential Power Line.* UbiComp 2007, 2007: p. 271–288.
14. Faloutsos, C., H. Jagadish, and N. Sidiropoulos, *Recovering Information from Summary Data.* VLDB'97, 1997: p. 36–45.
15. Brown, I., *An empirical comparison of benchmarking methods for economic stock time series*, U.C. Bureau, Editor. 2012.
16. Chen, B., *An empirical comparison of methods for temporal distribution and interpolation at the national accounts*, B.o.E. Analysis, Editor. 2007.
17. Sax , C. and P. Steiner, *Temporal disaggregation of time series.* The R Journal, 2013. **41**(5).
18. Chamberlin, G., *Temporal disaggregation.* Economic and Labour Market Review, 2010.
19. Gazzola, S. and J. Nagy, *Generalized arnoldi–tikhonov methodforsparsereconstruction.* SIAM Journalon Scientific Computing, 2014. **36**(2): p. B225–B247.

20. Golub, G., P. Hansen, and D. O'Leary, *Tikhonov regularization and total least squares.* SIAM Journal on Matrix Analysis and Applications, 1999. **21**(1): p. 185–194.

21. Liu, Z., et al., *H-FUSE: Efficient Fusion of Aggregated Historical Data.* Proceedings of the 17th SIAM International Conference on Data Mining, SDM 2017, 2017: p. 786 - 794.

22. Yang, F., et al., *Ares: Automatic Disaggregation of Historical Data.* VLDB, 2017.

23. Chester, U. and J. Ratsaby, *Universal distance measure for images.* IEEE 27-th Convention of Electrical and Electronics Engineers in Israel, 2012.

24. Marshall, M. and S. Gunasekaran, *A Survey on Image Retrieval Methods*, in *CIET-ECE DEPT*. 2014.

25. Clough, P. and M. Sanderson, *User experiments with the Eurovision crosslanguage image retrieval system.* Journal of the American Society of Information Science and Technology, 2006. **57**(5).

26. Eakins, J., *Towards Intelligent image retrieval' pattern recognition* Pattern Recognition, 2002.

27. Fauzi, M., *Low Quality Image Retrieval System For Generic Databases* Linear Networks and Systems, 2004.

28. Smeulders, A., M. Worring, and A. Gupta, *Content based image retrieval at the end of the early years.* IEEE Transactions on Pattern Analysis and Machine Intelligence 2000. **22**(12).

29. Lin, C., R. Chen, and Y. Chan, *A smart content-based image retrieval system based on color and texture feature.* Image and Vision Computing, 2009. **27**.

30. Kekre, H., et al., *Image retrieval using texture features extracted using LBG, KPE, KFCG, KMCG, KEVR with assorted Color spaces.* International Journal of Advances in Engineering & Technology, 2012. **2**(1).

31. Krishna, M. and S. Bhanu, *Content Bases Image Search And Retrieval Using Indexing By K Means Clustering Technique.* International Journal of Advanced Research in Computer and Communication Engineering, 2013. **2**(5).

32. Syam, B. and Y. Rao, *An effective similarity measure via genetic algorithm for content based image retrieval with extensive features.* International Arab journal information technology, 2013. **10**(2).

33. GALI, N., V. RAO, and A. SHAIK, *Color and Texture Features for Image Indexing and Retrieval* International Journal of Electronics Communication and Computer Engineering 2012. **3**(1).

34. Wijesekera, W. and W. Wijayanayake, *Low Quality Image Retrieval System For Generic Databases* International Journal of Scientific & Technology Research, 2015. **4**(12).

35. Chen, C. and H. Chu, *Similarity Measurement Between Images.* Computer Software and Applications Conference, 2005. COMPSAC 2005. 29th Annual International, 2005. **2**.

36. Wang, Z., et al., *Image quality assessment: From error visibility to structural similarity.* IEEE Transactions on Image Processing, 2004. **13**(4).

37. Dosselmann, R. and X. Yang, *A Comprehensive Assessment of the Structural Similarity Index.* Signal Image and Video Processing 2009. **5**(1).

38. Kondylidis, N., M. Tzelepi, and A. Tefas, *Exploiting tf-idf in deep convolutional neural networks for content based image retrieval.* MultimediaToolsandApplication, 2018. **77**(33).

39. Fawcett, T., *An Introduction to ROC Analysis.* Pattern Recognition Letters, 2006. **27**(8): p. 861–874.

40. Powers, D., *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation.* Journal of Machine Learning Technologies, 2011. **2**(1): p. 37-63.

41. Sara, U., M. Akter, and M. Uddin, *Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study*. Journal of Computer and Communications, 2018. **7**(3).

42. *The CIFAR-10 dataset*, in https://www.cs.toronto.edu/~kriz/cifar.html.

43. Keogh, E., J. Lin, and W. Truppel, *Clustering of time series subsequences is meaningless: Implications for previous and future research*, in *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*. 2003, IEEE Computer Society: Washington, DC, USA.

44. !!! INVALID CITATION !!! {}.

45. Kelvin. and M. Wong, *Fast time-series searching with scaling and shifting*, in *PODS '99: Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 1999, ACM: New York, NY, USA.

46. Myers, C., L. Rabiner, and A. Rosenberg, *Performance tradeoffs in dynamic time warping algorithms for isolated word recognition.* IEEE Transactions on Signal Processing, 1980.

47. Yazdani, N. and M. Ozsoyoglu. *Sequence matching of images*. in *SSDBM '96: Proceedings of the Eighth International Conference on Scientific and Statistical Database Management*. 1996. Washington, DC, USA: IEEE Computer Society.

48. Cartwright, K., P. Russell, and E. Kaminsky, *Finding the maximum and minimum magnitude responses (gains) of third-order filters without calculus.* Lat. Am. J. Phys. Educ, 2013. **7**(4).

49. Chaovalit, P., et al., *Discrete Wavelet Transform-Based Time Series Analysis and Mining.* ACM Computing Surveys, 2011. **43**(2).