International Journal of Research in Advent Technology, Special Issue, August 2018
E-ISSN: 2321-9637
International Conference on "Topical Transcends in Science, Technology and Management"
(ICTTSTM-2018)
Available online at www.ijrat.org

An Approach for Cover Song Identification through a Bunch of Upcoming Optimal Classifiers

D. KHASIM VALI[1],
Department of computer Science and Engineering, Vidyavardhaka College of Engineering, Mysuru – 570017,
e-mail: dkv@vvce.ac.in
NAGAPPA U BHAJANTRI[2],
Department of computer Science and Engineering, Government College of Engineering, Chamarajanagara – 571313, e-mail: bhajan3nu@gmail.com.

ABSTRACT

The identification of a cover song, which is an alternative version of a previously recorded song for music retrieval, has received further attention. However, there is an urgency to protect the interest of melody music contributors. In view of this, several efforts have been heaping in literature. In other words, few efforts milestone in this direction [1]. Methods for identifying a cover song typically involved in comparing the similarity of chroma features between a query and another song. Moreover, considerable amount of time is essential for pairwise comparisons.

In this effort, chroma features are patched to preserve the melody. An intermediate representation like Discrete Cosine Transform (DCT), Fast Fourier Transform (FFT) and Discrete wavelet transform (DWT) are trained to reduce the dimension of each patch of chroma features. The training was performed using Support Vector Machine (SVM), Naïve Bayes and K-Nearest Neighbor (KNN). Further optimized kernel has exercised. Interestingly SVM optimal kernel has shown remarkable improvement with the cover song dataset. The Experimental results showcased the proposed method potentiality by its capability. In other words, noticed the better accuracy for cover song identification and similarity matching on covers30 and covers80 dataset.

Keyword: KNN, SVM, DCT, FFT, DWT, Chroma

I. INTRODUCTION

A cover song, or simply cover, is a new version of existing music recorded or arranged by another musician. A cover reuses the melody and lyrics of the original song, but it is performed with new singers and instruments. Further musical factors such as key, rhythm, and genre can be reinterpreted by the new artist. Since the copyright of composition of the cover still belongs to the author of the original song, releasing a cover song without permission of the genuine contributor may cause a legal conflict.

Another case is music sampling, which is the act of process that reuses a snippet of existing sound recordings. The sampling is widely considered to be a technique for retrieving music today, but licensing that the original creator authorizesits reuse is a legal requirement. In other words, Cover song identification is a task that aims to measure the similarity between two songs. It can be used to prevent the infringement of copyright, and also to be an objective reference in case of conflict.

For a decade many approaches for cover song identification have been proposed [9, 17, 19]. Humans generally recognize the cover through the melodic or lyric similarity, but separation of the predominant melody from a mixed music signal is still not at a reliable level, and extraction of the lyrics can be attempted only if it is clearly separated.

The created dataset "covers30", contains 30 sets of original and cover songs - spanning across genres, styles, live and recorded music, it is biased towards regional languages. Most of songs contain a cover version however some of songs contain up to three. Similarly, the extended "covers80" [14] employed at MIREX 2007 to benchmark cover song recognition systems. The dataset contains 80 sets of original and cover songs, 166 in total –which comprises genres, styles, live and recorded music. In other words covers80 predominantly, oriented towards western music.

Rest of the paper is organized in the following way. In section II we present a deep coverage on broader area of cover song identification. The emphasize of proposal strategy is in section III.Section IV dedicated to comment on chroma features extraction through reduction criteria. Further, improvement of kernel has articulated via optimization and same is portrayed in section V. Outcomes are quantified with a yardstickviaempirically as well as statistically narrated and experimental validations are presented in section VI. Conclusions are drawn in section VII.

II. RELATED WORK

Over the last years, in the area of cover song identification, there has been a considerable amount of new approaches and techniques that try to handle different issues [2, 4, 14]. The typical goal is to try new algorithms or combinations of them in order to improve the results in comparison to previous systems, but the recent main focus by most researchers has been towards scalable strategies [21, 22].The most common way to calculate the similarity between two different songs is through the use of alignment-based methods and they have shown to produce good results, 75% MAP in MIREX'2009[1].

However, these methods are computational expensive and, when applied to large databases, they can become impractical: the best performing algorithm [1] in MIREX'2008 implemented a modified version of the Smith-Waterman algorithm and taken approximately 104 hours to compute the results for 1,000 songs. If algorithm applied to the Million Song Dataset (MSD), the estimated time to conclude would be of 6 years [2]. Martin et al. [3] suggest the use of Basic Local Alignment Search Tool (BLAST), a bioinformatics sequence searching algorithm, as an alternative to dynamic programming solutions.

The data is indexed based in similarity between songs, and to compute the similarity value, the best subsequences are chosen, and then compared. Khadkevich et al. [4] extract information about chords and store them using Locality-Sensitive Hashing (LSH). Bertin-Mahieux et al. [3] adopt the2D Fourier Transform Magnitude for large-scale cover detection. This solution was further improved by Humphrey et al. [5], who modified the original work to use a sparse, high-dimensional data-driven component, and a supervised reduction of dimensions.

The authors J.V. Balen et al. [2] extract high-level musical features that describe harmony, melody, and rhythm of a musical piece. Those descriptors are stored with LSH, which allows retrieving the most similar songs. Lu and Cabrera [6] use hierarchical K-means clustering on chroma features to find audio words, in other words centroids. A song will then be represented by its audio words. Moreover similarity with other songs will be determined by audio words share with the same location.

Outside the field of large-scale cover identification, several solutions regarding distance fusion[4] have been suggested. Salamon et al. [7] extract the melodic line, the bass-line, and HPCP 12-bins for each song. They explore the fusion of those features in order to discover which results in the best performance. Distance fusion is also the main focus in the work of Degani et al. [6], where they propose a heuristic for distance fusion. Their proposal consists of normalizing all values to [0, 1], computing a refined distance value, and produce a single matrix of results.

The conventional approaches described above calculate the distance between a query and the songs to be compared, and determine that the song with the nearest distance is highly likely to be a cover. Since this process is separate from each query, the result from "another version of the same cover" cannot be taken into account. If it is possible, songs with different lengths can be represented in the same space.

Furthermore, if similar/dissimilar song pairs are known, the metric to measure the song distance can be optimized, rather than using the Euclidean distance. Instead of taking the distance matrix directly to rank the similarity, here first perform a transformation using DCT to rearrange each song in the high-dimensional space. Next, the distance metric is learned from song pairs in the new representation and their labels. Similarly select "core songs" with diverse musical properties and use them for both embedding and training. In summary, the approach assumes that the distance between the core set and each song can be a discriminating feature to easily group the same covers. The conceptual illustration of this new representation is shown in Figure 1.

The goal of this work is to examine whether the bunch of optimal classifiers can be effectively retrieve the similarity between songs. Also, aims to achieve the best performance in cover song identification by applying the DCT to the matrix generated which was comprehended by existing algorithms [14].

III. PROPOSED METHOD

The proposed system, shown in Figure 1, portrayed in three stages. Firstly, the preprocessing stage, attempts to convert audio signals into chroma features for each song. Then reduce the chroma features through DCT, FFT and DWT coefficients.Subsequently diminished features will fed into various classifiers such as SVM, NB and KNN.

A. Beat Tracking

Since erected approach basic representation consists ofa feature vector per beat, it must start by identifying the beat segmentation times in the music audio. The first stage of beat tracking converts the audio into an "onset strength" value at a 250 Hz sampling rate. On the other hand, this is derived by taking the first-order difference along time in a log-magnitude 40-channel Mel-frequency spectrogram, throwing away negative values, then summing across frequency. Further slowly-varying offsetsare removed by a high pass filter at about 0.5 Hz.

Then, an approximate global tempo is estimated by auto-correlating the onset strength, applying a 'preference window' which is a Gaussian on a log-time axis, and choosing the period with the largest windowed autocorrelation as the tempo. Similarly, variation of the center of the preference window between 0.25s and 0.5s i.e. between 240 and 120 beats per minute (BPM) is employed to obtain beat segments at different points in the metrical hierarchy of the music. Consequently employed dynamic programming to find the set of beat times that optimizes both the onset strength at each beat and the spacing between beats.

On the other hand dynamic programming is an efficient way to search all possible beat sequences to optimize a total cost that can be broken down into a local score at each beat time, and a transition cost. Conventionally, the transition cost is additive, but here maid an attempt to implement it as a scaling window, again a Gaussian on a log-time axis, applied to the onset strength envelope for 0.5 . . . 2.0 the tempo period prior to the current time, with the maximum at the target period. For every possible beat time, the best preceding beat time is located as the maximum of the scaled onset strength within the window, and the cumulative score up to that beat is calculated. Then, the largest score close to the end of the audio is located, and the entire sequence of beats leading to that beat time is recovered through a 'backtrace' table storing the predecessor for every beat time.

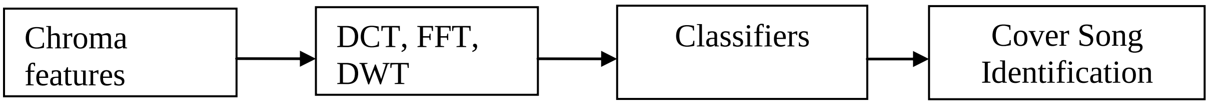


Figure 1: The proposed method

B. Chroma Features

If the beat tracking can identify the same main pulse in different renditions of the same piece, then representing the audio against a time base defined by the detected beats normalizes away variations in tempo. Further choose to record a single feature vector per beat, and use twelve element 'chroma' features to capture both the dominant note, typically melody as well as the broad harmonic accompaniment [8, 9].Hence the Chroma features record the intensity associated with each of the twelve semitones for example piano keys within one octave, but all octaves are folded together. The idea of calculating harmonic features over beat-length segments appears in [10].

Instead of using a coarse mapping of FFT bins to the chroma classes they overlap, use the phase-derivative within each FFT bin both to identify strong tonal components in the spectrumand to get a higher-resolution estimate of the underlying frequency [11, 12]. Further this technique assist to remove non-tonal components and improve frequency resolution beyond FFT bin level has similar motivation and impact to the sinusoid-modeling-based preprocessing proposed by [13]. By using only components up to 1 kHz chroma features worked suitably.

IV. REDUCTION METHODS

A. Discrete Cosine Transforms

We normalized Chroma features using DCT, it is widely employed for image compression [39] and dimensionality reduction of feature data [39]. A DCT expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies. The DCT is conceptually similar to the discrete Fourier transform (DFT), but DCT does a better job than DFT by concentrating energy into lower order coefficients for feature data. A DCT operation on a feature data produces coefficients that are similar to the frequency domain coefficients produced by a DFT operation.

An N-point DCT has the same frequency resolution as and is closely related to a 2N-point DFT. The N frequencies of a 2N point DFT correspond to N points on the upper half of the unit circle in the complex frequency plane. For most feature data, after transformation the majority of signal energy is carried by just a few of the low order DCT coefficients. These coefficients can be more finely quantized than the higher order coefficients. Many higher order coefficients may be quantized to 0 more precisely which allows for very efficient run-level coding. The efficiency purpose feature reduction varies from 100 to 400, and each feature is represented in symbolic.

B. Discrete Wavelet Transformation

The wavelet transform describes a multi-resolution decomposition process in terms of expansion of a signal onto a set of wavelet basis functions. DWT has its own excellent space frequency localization property. Application of DWT in 1D signal corresponds to 1D filter in each dimension. Similarly, one such wavelet called Daubechies (db) employed as mother wavelet. Further, it is divided into eight non-overlapping multi-resolution sub-bands by the filters, namely db1, db2, db3up to db8. The sub-band is processed further to obtain the next coarser scale of wavelet coefficients, until some final scale "N" is reached. When a signal is decomposed into eight levels, the db6 sub-band signal best reflects the original signal [22,23].

C. Fast Fourier Transform

A Fast Fourier Transform (FFT) [21] is a faster, computational, efficient algorithm to compute the discreteFourier transform (DFT) [24] and inverse of DFT. Let $x_0, x_{1.........,} x_{N-1}$ be complex numbers and DFT is defined by the equation (1).

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} nk} k = 0,....,N-1. \quad \text{--------(1)}$$

Here, DFT is the transformation of the discrete signal taking in time domain into its discrete frequency domain representation[6].The transform and inverse transform pair given for vectors of length N via equation (2) and equation (3).

$$X(k) = \sum_{j=1}^{N} x(j)\omega_N^{(j-1)(k-1)} \quad \text{--------(2)}$$

$$x(j) = (1/N)\sum_{k=1}^{N} X(k)\omega_N^{-(j-1)(k-1)} \quad \text{--------(3)}$$

The FFT algorithm reduces the computational burden to O(nlog n) arithmetic operation. FFT requires the number of data points to be a power of 2 and requires evenly spaced time series.

V. CLASSIFICATION

The reduced chroma features using DCT, DWT and FFT are fed into KNN, NB, SVM and optimized SVM classifiers. While brief introductions of these classifiers are given in the following subsections.

A. K- Nearest Neighbor (KNN)

One of the simplest classifier amongst all the classifiers is the K-Nearest Neighbor classifier [15, 16]. The term K-nearest can be interpreted as the k-number of points which are relatively closer with respect to a distance to a point in n-dimensional feature space. In concern with this, the Euclidean distance is computed between a test and the train samples. The k-training samples with relatively less distances are termed as K-Nearest Neighbors. So, among k-training samples, the maximum number of samples which are most similar to a test, proves out to allocate its class label to the given test sample. This exploits the 'smoothness' assumption that samples close to each other will probably have the same class and in our work, we considered K as 1.

B. Support vector machine

The support vector machine (SVM) is superior to all machine learning algorithms which are based on statistical learning theory. There are a number of publications detailing the mathematical formulation and algorithm development of the SVM [19, 20]. The inductive principle behind SVM is structural risk minimization (SRM), which constructs a hyper-plane between two classes, such that the distance between support vectors to the hyper-plane would be maximum. In order to deal with non-linearly separable classes, the input data are first mapped using a kernel to a higher dimensional space in SVM. Further, the radial basis function (RBF) kernel is extended in SVM.

C. Optimal Support vector machine

A critical step in support vector machine classification is choosing a suitable kernel of SVMs for a particular application, i.e. various applications are need of different kernels to get reliable classification results. Moreover, it is well known that the two typical kernel functions often employed in SVMs are the radial basis function kernel and polynomial kernel. Further, more recent kernels are presented in [26, 27, 28, 29, 30]tohandle high dimension data sets and are computationally efficient when handling non-separable data with multi attributes.

However, it is difficult to find kernels that are able to achieve high classification accuracy for a diversified data set. In order to construct kernel functions from existing ones also some other simpler kernel functions as building blocks, the closure properties of kernel functions are essential [32,33].

For given non-separable data, in order to be linearly separable, a suitable kernel has to be chosen. Classical kernels, such as Gauss RBF and POLY functions, can be used to transfer non-separable data to separable, but their performance in terms of accuracy is dependent on the given data sets. The following POLY function performs well with nearly all data sets, except high dimension ones [32]:

$$\text{POLY}(x,y)=(x^{T}z+1)^{d} \quad \text{--------}(4)$$

Where *d* is the polynomial degree. The same performance is obtained with the Gauss RBF of the following form[28]:

$$\text{RBF}(x,z)= \exp(-\gamma\|x-z\|^{2}) \text{ -------}(5)$$

Where γ is appositive parameter controlling of the radius. Zanaty et.al.,[26] in presented the polynomial radial basis function (PRBF) as:

$$\text{PRBF}= ((1+\exp(\omega))/V)^{d} \text{ -------}(6)$$

where x = |x z|, V = p * d and p is a prescribed parameter. Completely achieving a SVM with high accuracy classification therefore, requires specifying high quality kernel function.

Here, we are amalgamating POLY, RBF, and PRBF into a kernel to become:

$$\text{GRPF}(x,z)=\left(\frac{d+r.\exp(-\|X-z\|^{r}/(r-\sigma^{2}))}{r+d}\right)^{d+1} \qquad \text{--------}(7)$$

where σ is a statistic distribution of the probability density function of the input data; and the values of *r(r > 1)* and d can be obtained by optimizing the parameters using the training data. The proposed kernel has the advantages of generality.

However, the existing kernels such as PRBF in Zanaty et al. [26], Gaussian and polynomials kernel function by setting ***d*** and ***r*** in different values. For example if *d* as *0*, we get Exponential Radial when *r* as 1 and Gaussian Radial for *r* as *2* and so on. Moreover various kernels can be obtained by optimizing the parameters using the training data.

Further, an effort has continued to produce better classification results via optimized kernel of SVM and it is revealing the evidences.

D. Optimizing the kernel parameters

The SVM algorithm emphasizes that the kernel GRPF depends on two parameters *d and r,* encoded into a vector θ as*(d, r).* Thus consider a class of decision functions parameterized by α , *b,* and θ :

$$f_{\alpha,b,\theta}(x)=sign(\sum_{i=1}^{l}\alpha_{i}y_{i}GRPF_{\alpha}(x,z)+b) \text{ -------}(8)$$

More precisely, for θ fixed, we want to have α^{0} = arg max w(α) and choose θ^{0} such that:

$$\theta^{0}=\arg\min_{\theta} T(\alpha^{0},\theta) \text{ -------}(9)$$

When, θ is a one dimensional parameter, one typically tries a finite number of values and picks the one which gives the lowest value of the criterion T. When both T and the SVM solution are continuous with respect to θ a better approach has erected by Cristianini et al. [34].Author employed an incremental optimization algorithm, one can train an SVM with little effort when θ is changed by a small amount.

However, as soon as θ has more than one component computing T(α , θ) for every possible value of θ becomes intractable, and one rather looks for a way to optimize T along a trajectory in the kernel parameter space.

In this work, the gradient of a model selection criterion to optimize the model parameters [35,36]has extended. This can be achieved by the procedure following iterative steps:

1. Initialize θ to some value.
2. Extended SVM to find the maximum of the quadratic form α^{0} = arg max w(α).
3. Renovate θ such that T is minimized. This is typically achieved by a gradient step Chapelle et al. [38].
4. Repeat until T is minimized.

Solving step 3 requires estimating how T varies with θ where GRPF θ can be differentiated with respect to θ [38].In order to evaluate the performance of the SVM with different kernels carried out some experiments with different data sets.

E. Naive Bayes

Naive Bayes [19] is a successful classifier based upon the principle of Maximum a Posteriori (MAP). Given a problem with K classes $\{C_1, \ldots, C_K\}$ with prior probabilities $P(C_1), \ldots, P(C_K)$, we can assign the class label c to an unknown example with features $x = (x_1, \ldots, x_N)$ such that c = argmaxcP(C = c||$x_1, \ldots, x_N$), that is choose the class with the maximum a posterior probability given the observed data.

This aposterior probability can be formulated, using Bayes theorem, as follows: P(C = c||$x_1, \ldots, x_N$) = P (C=c)P ($x_1,\ldots,x_N$|| C=c) \ P ($x_1,\ldots,x_N$) . As the denominator is the same for all classes, it can be dropped from the comparison. Now, compute the class conditional probabilities of the features given the available classes. This can be quite difficult taking into account the dependencies between features. The naive Bayes approach is to assume class conditional independence i.e. $x_1, \ldots, x_N$ are independent given the class.

VI. EXPERIMENTAL VALIDATIONS

We have experimented on created and real datasets in order to reveal the capability of proposed criteria. This method has been implemented in a Matlab R2013a using an Intel Pentium 4 processor, 2.99 GHz Windows PC with 1 GB of RAM.

In this work, an attempt is made to compute DCT, FFT and DWT coefficients as a dominant feature with different datasets like covers30 and covers80. In addition, to reveal performances of different classifiers like SVM, KNN and NB by varying coefficients from 100 to 400, the same is displayed in Table 1 and 2.

The process explicitly dedicated to estimate the dominant features of cover song has been continued. The estimated method of DCT based features has shown significant change in the classifiers outcome.

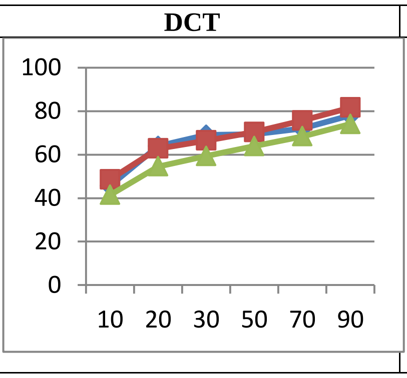
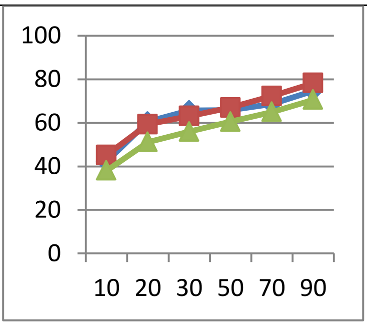
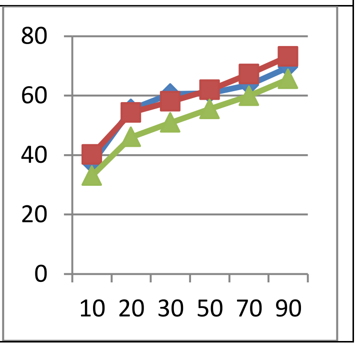
However, normal kernel function has remarked some potentials and hence it is further motivated to mine itthrough the proposed novel criteria, such as the optimal kernel of SVM has been employed to estimate and classify the cover songs.

On the other hand, improved effort has extended through optimal SVM by tuning the parameters of normal SVM which is called as kernel GRPF. The experimental outcomes have shown in Table 3 and 4.

The obtained results witnessed through maximum accuracy remarked in all cases. In view of subsequent appreciation and corroborated the supremacy of optimized kernel, the experimentation has been explored on database under varying size from 10 to 90 percent of database for all the cases.

The proposed effort has employed the following metrics like empirical, statistical and hybrid that is mixture of songs to evaluate the performance potential.

Table 1: Classification accuracy for various Classifiers on Cover 30

	DCT	FFT	DWT
100			

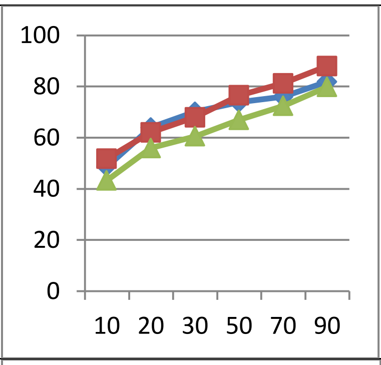
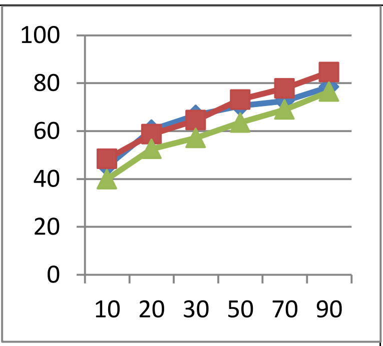
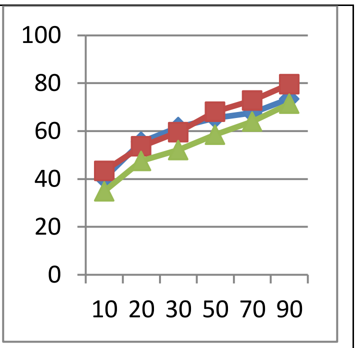
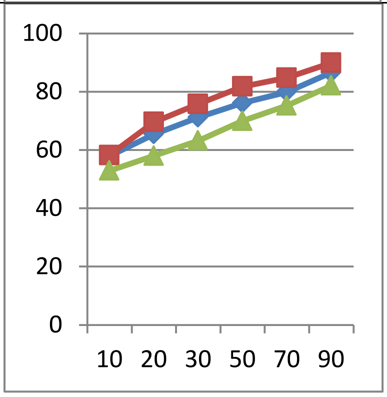
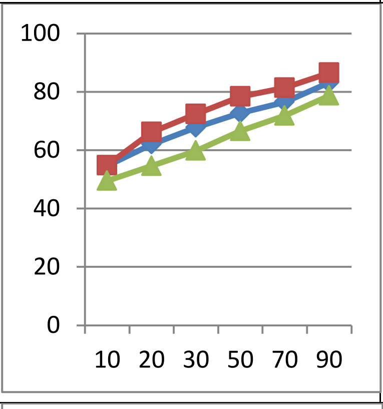
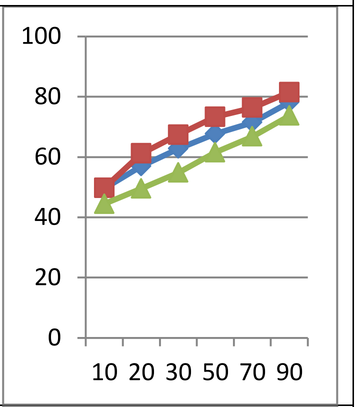
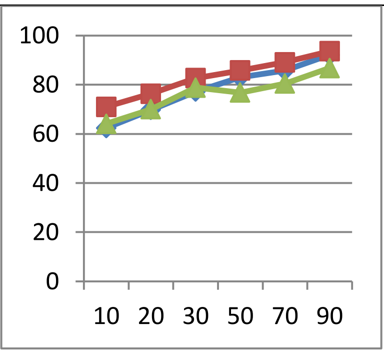
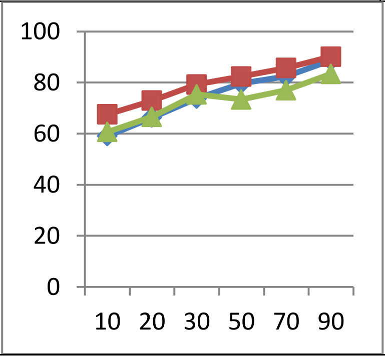
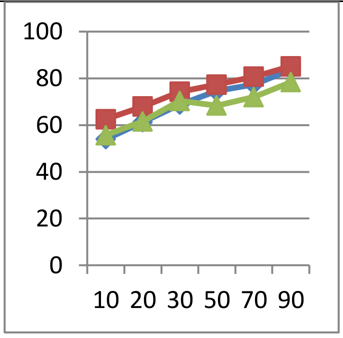
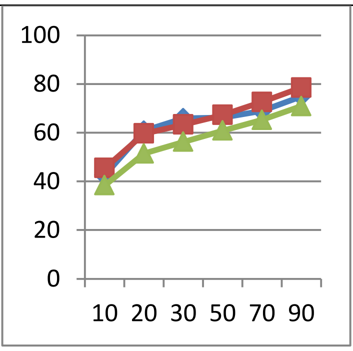
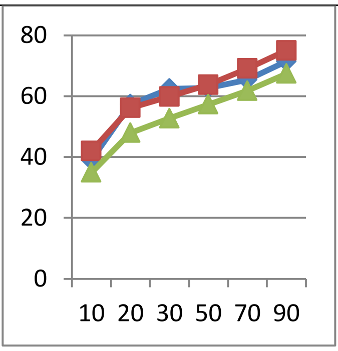
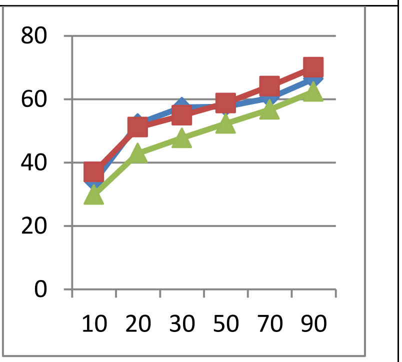
200			
300			
400			

Table 2: Classification accuracy for various Classifiers on Cover 80

	DCT	FFT	DWT
100			

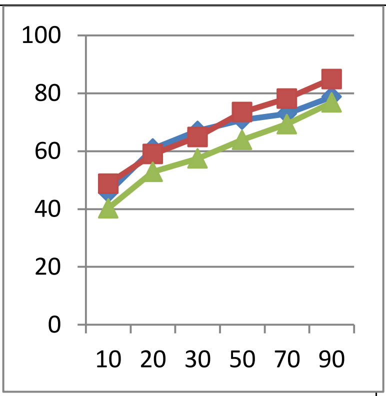
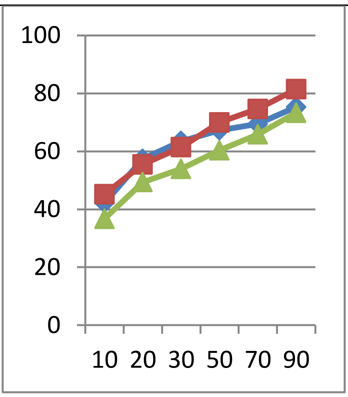
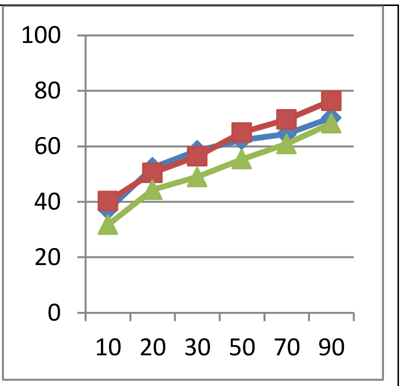
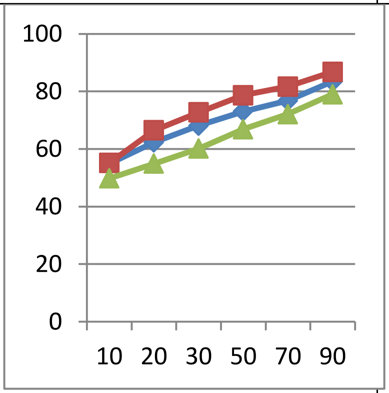
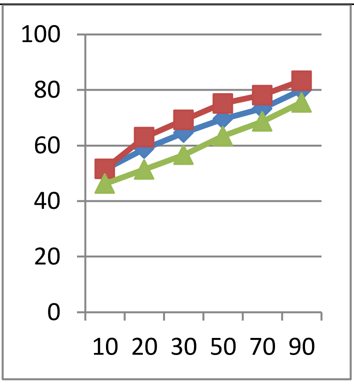
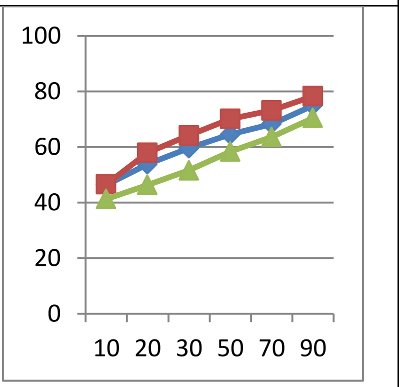
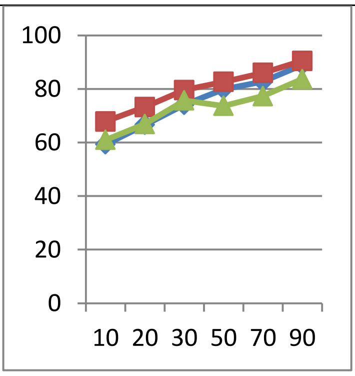
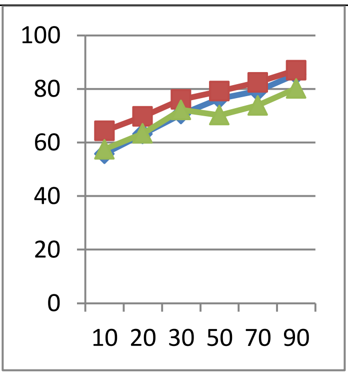
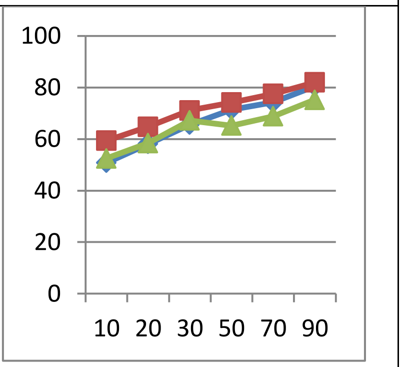
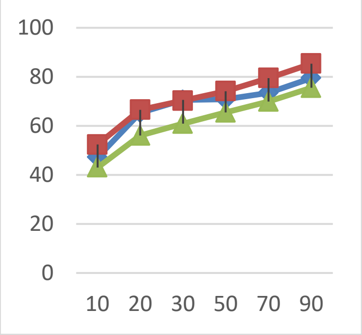
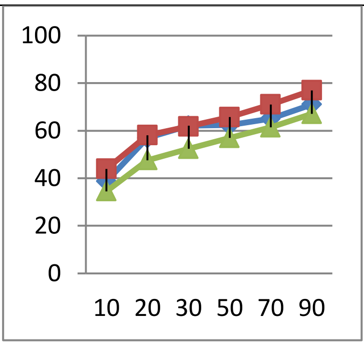
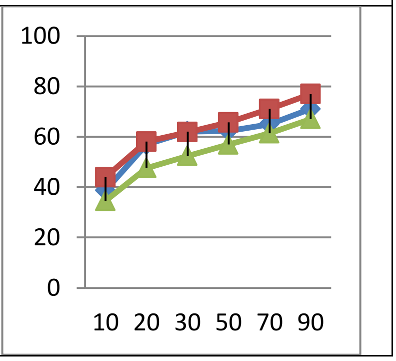
200			
300			
400			

Table 3: Classification accuracy for various optimized Classifiers on Cover 30

	DCT	FFT	DWT
100			

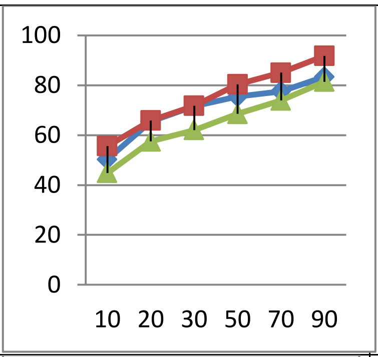
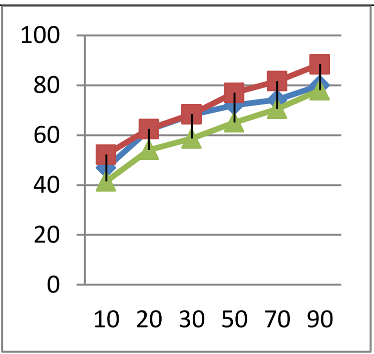
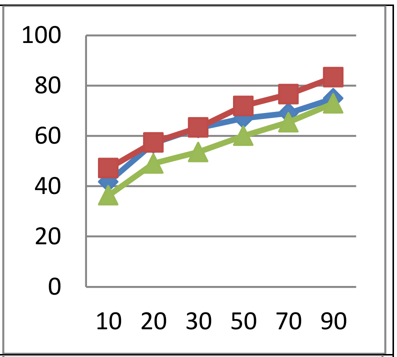
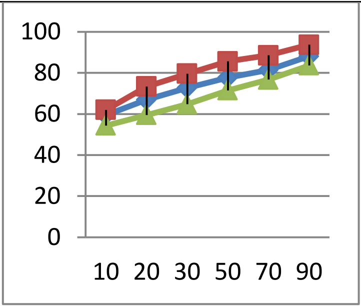
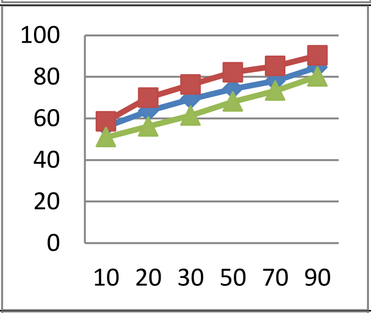
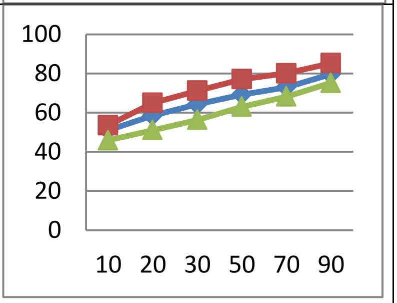
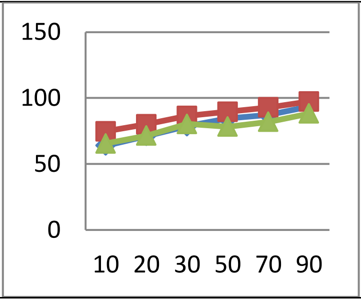
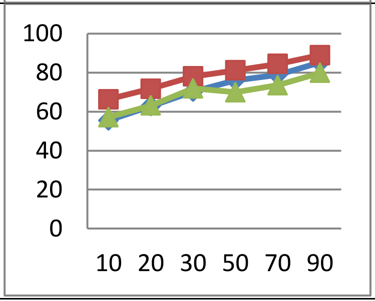
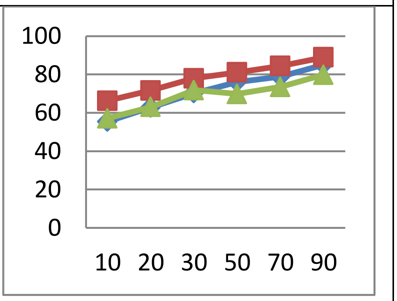
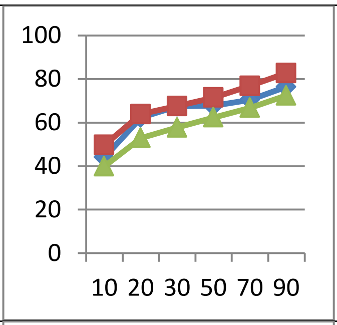
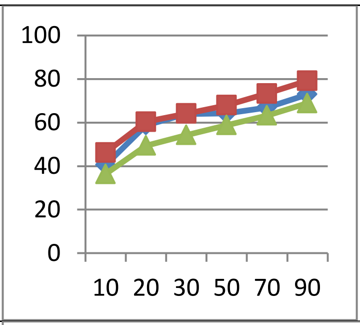
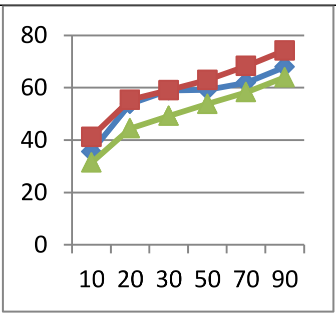
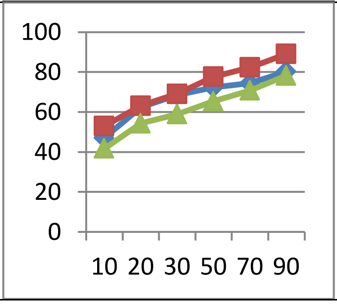
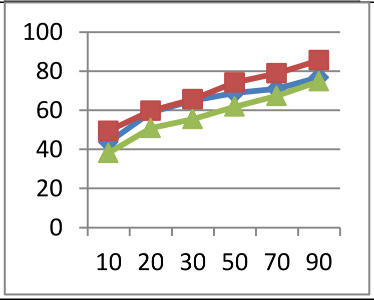
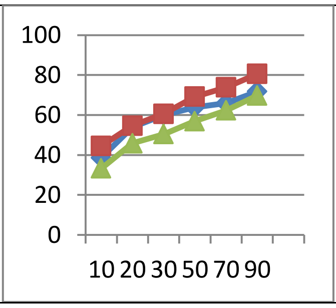
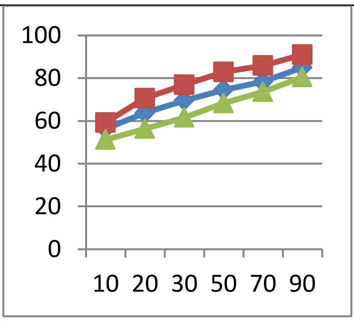
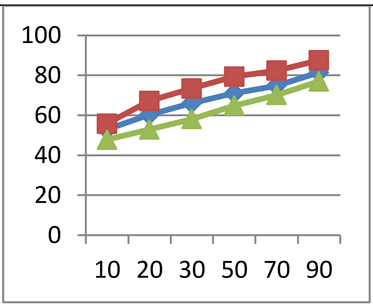
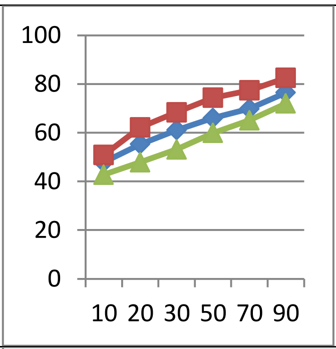
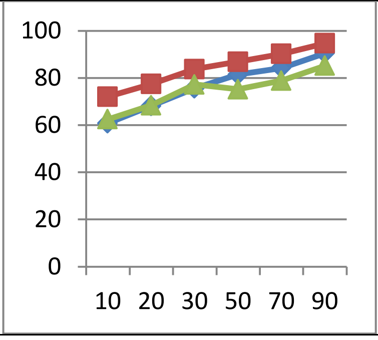
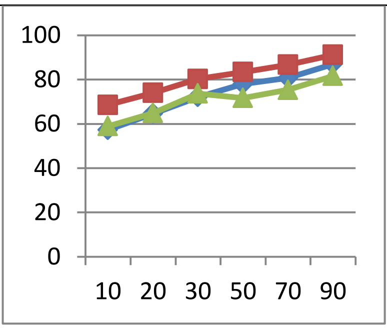
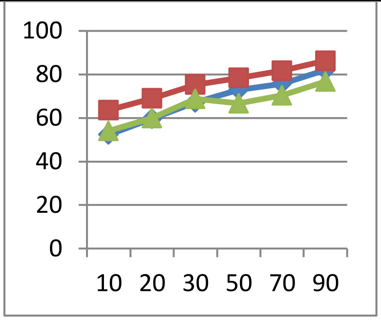
200			
300			
400			

Table 4: Classification accuracy for various optimized Classifiers on Cover 80

	DCT	FFT	DWT
100			
200			

300			
400			

A. Empirical metrics:

Now, SVM kernel amplification has force on convinced empirical performance benchmarks put into effect. However, accuracy increases linearly as more coefficients heaped.

Further, to strengthen the performance measurement has achieved through improved kernel. The SVM, Optimized SVM on covers 30 and 80 dataset using DCT is illustrated via bar charts in Figure 2 and 3 respectively. As coefficients increase accuracy enhanced linearly and size of dataset increase accuracy diminishes linearly.

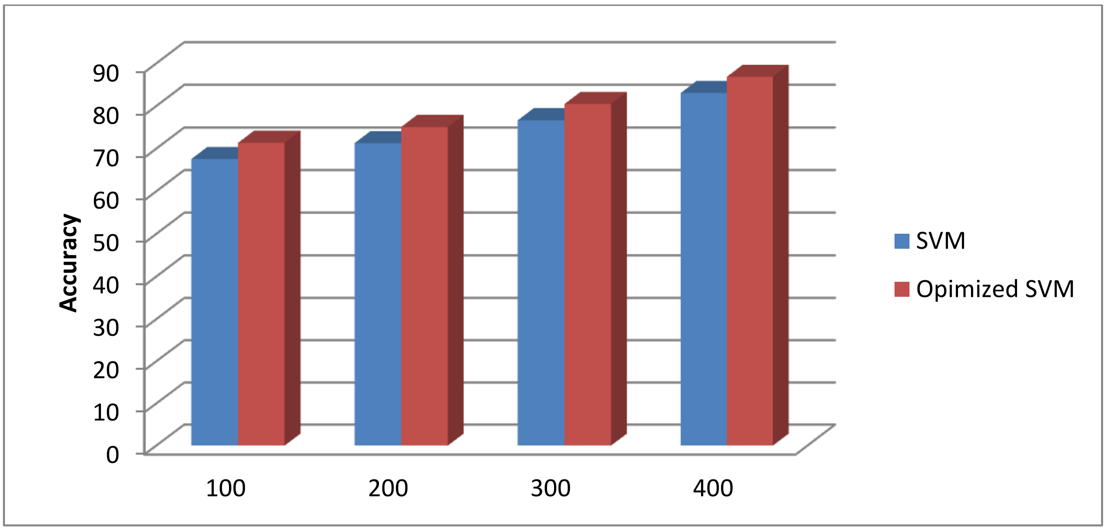


Figure 2: Variation between SVM and Optimized SVM on covers 30 dataset using DCT

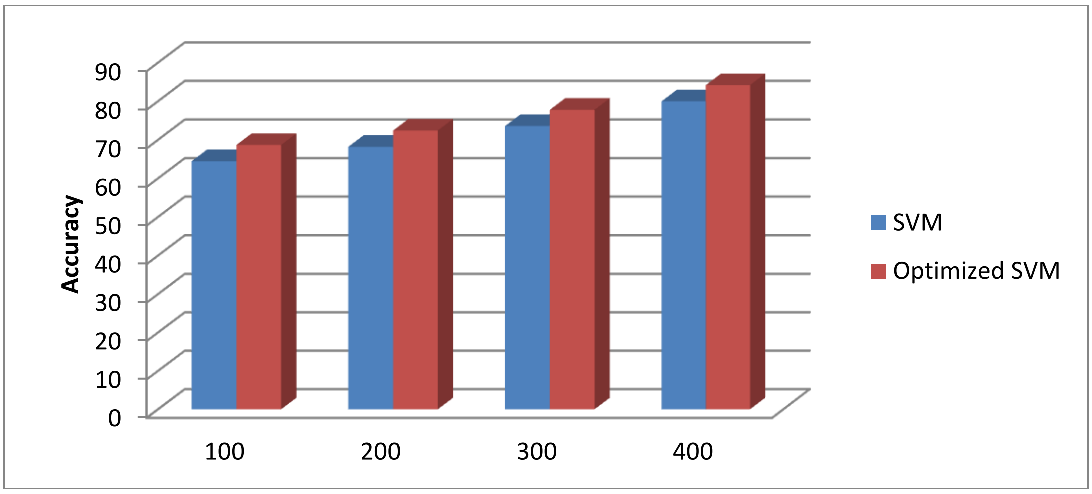


Figure 3: Variation between SVM and Optimized SVM on Covers 80 dataset using DCT

B. Statistical metrics:

At this juncture, kernel enhancement has impact on certain statistical performance yardsticks. In other words, accuracy increases in percentage from 91 to 92, whereas precision advances from 94 to 95, prevalence progresses from 82 to 83 for cover 30. Correspondingly, accuracy enlarges from 90 to 92.7, while precision raise from 90 to 91, prevalence stay unchanged for covers 80 with DCT 400 coefficients, which is revealed in Table 5, 6, 7 and 8. In the same way, 100,200 and 300 coefficients exhibited the greatly analogous consequences.

There are two possible predicted classes: "yes" and "no". The classifier made a total of 166 predictions out of those, the classifier predicted "yes" 111 times, and "no" 55 times. Table 7 and 8 shows the confusion matrix of Cover song identification for covers 80. The table 7 and 8 highlight the precision and accuracy about 90%. Similarly the experimentation is conducted on covers 30 and obtained accuracy is about 91%. The Confusion matrix is tabulated in table 5 and 6.

Table 5: Confusion Matrix of Cover song identification for DCT-400 in Covers 30 with Normal Kernel.

metrics	N=80	Predicted		Accuracy: (TP+TN)/total	Misclassification Rate (FP+FN)/ Total	Precision: TP/ predicted yes	Prevalence: Actual yes/ Total
		Predicted: NO	Predicted: YES				
Actual	NO	TN=10	FP=4	0.91	0.08	0.94	0.82
	YES	FN=3	TP=63				

Table 6: Confusion Matrix of Cover song identification for DCT-400 in Covers 30 with optimal Kernel.

metrics	N=80	Predicted		Accuracy: (TP+TN)/ Total	Misclassification Rate (FP+FN)/ total	Precision: TP/ predicted yes	Prevalence: Actual yes/total
		Predicted: NO	Predicted: YES				
Actual	NO	TN=10	FP=3	0.92	0.06	0.95	0.83
	YES	FN=2	TP=65				

Table 7: Confusion Matrix of Cover song identification for DCT-400 in Covers 80 with Normal Kernel.

metrics	N=166	Predicted		Accuracy: (TP+TN)/total	Misclassification Rate (FP+FN)/total	Precision: TP/ predicted yes	Prevalence: Actual yes/total
		NO	YES				

Actual	NO	TN=50	FP=10	0.90	0.09	0.90	0.64
	YES	FN=5	TP=101				

Table 8: Confusion Matrix of Cover song identification for DCT-400 in Covers 80 with optimal Kernel.

metrics	N=166	Predicted		Accuracy: (TP+TN)/ total	Misclassification Rate (FP+FN)/ Total	Precision: TP/ predicted yes	Prevalence: Actual yes/total
		NO	YES				
Actual	NO	TN=50	FP=10	0.927	0.07	0.91	0.64
	YES	FN=2	TP=104				

C. Mixture of Original Song (OS) and Cover Song (CS):

Here, enhanced observation of results for appropriate mixture of OS and CS has remarked through table 9.However, simplified kernel of SVM has witnessed the accuracy domination among other two classifiers such as NB and KNN. The exclusive extension of improved SVM also continued the performance. Hence, the experiment has exhibiting and supplementing the emphasize of outcomes. The results are tabulated in table 9,itcan be noticed that numerical results are more encouraging for 50 samples with DCT 400.

Table 9: Testing results of OS versus CS for DCT-400 in Cover30 and Cover80

Number of Samples for testing	Datasets		Normal Classifier			Optimal Classifier		
			NB	KNN	SVM	NB	KNN	SVM
50	Cover 30	CS	74.16	77.47	81.55	79.16	82.47	86.97
		OS and CS	73.46	80.85	85.86	82.66	85.97	90.47
	Cover 80	CS	74.23	75.16	78.66	79.84	80.77	84.97
		OS and CS	77.98	78.63	82.14	82.00	82.49	86.77

Thus, the yardsticks of empirical, statistical and mixture dataset outcomes are exercised and revealed that improvised SVM with DCT has glimpsed its supremacy of modality.

VII. Conclusion

Here, proposed work has exercised on Covers30 and 80, which is enabling to reveal noticeable outcomes. Further, effort initiated to extract the spectral coefficients as dominant features to represent the identical tune based song, especially via DCT coefficients. Subsequently, classifiers are enrolling to categorize them; majorly optimized kernel of SVM has demonstrated its potential with noticeable upshot.

REFERENCES

[1] J. Serr`a, E. G´omez, P. Herrera, and X. Serra. Chroma binary similarity and local alignment applied to cover song identification. IEEE Transactions on Audio, Speech and Language Processing, 16:1138–1151, 08/2008 2008.

[2] J. V. Balen, D. Bountouridis, F. Wiering, and R. Veltkamp. Cognition-Inspired Descriptors for Scalable Cover Song Retrieval. In ISMIR'14, pages 379–384, 2014.

[3] B. Martin, D. G. Brown, P. Hanna, and P. Ferraro. BLAST for Audio Sequences Alignment: A Fast Scalable Cover Identification Tool. In ISMIR'12, number Ismir, pages 529–534, 2012.

[4] M. Khadkevich and M. Omologo. Large-Scale Cover Song Identification Using Chord Profiles. In ISMIR'13, pages 233–238, 2013.

[5] Y. Lu and J. E. Cabrera. Large scale similar song retrieval using beat-aligned chroma patch codebook with location verification. In SIGMAP'12, pages 208–214, 2012.

[6] E. J. Humphrey, O. Nieto, and J. P. Bello. Data Driven and Discriminative Projections for Large-Scale Cover Song Identification. In ISMIR'13, pages 149–154, 2013.

[7] J. Salamon, J. Serr`a, and E. G´omez. Tonal representations for music retrieval: From version identification to query-by-humming. International Journal of Multimedia Information Retrieval, special issue on Hybrid Music Information Retrieval, 2:45–58, 2013.

[8] A. Degani, M. Dalai, R. Leonardi, and P. Migliorati. A Heuristic for Distance Fusion in Cover Song Identification. In WIAMIS'13, pages 1–4, 2013.

[9] T. Fujishima. Realtime chord recognition of musical sound: A system using common lisp music. In Proc. ICMC, pages 464–467, Beijing, 1999.

[10] M. A. Bartsch and G. H. Wakefield. To catch a chorus: Using chroma-based representations for audio thumbnailing. In Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, Mohonk, New York, October 2001.

[11] N. C. Maddage, C. Xu, M. S. Kankanhalli, and X. Shao. Content-based music structure analysis with applications to music semantics understanding. In Proc. ACM MultiMedia, pages 112–119, New York NY, 2004.

[12] F. J. Charpentier. Pitch detection using the short-term phase spectrum. In Proc. ICASSP-86, pages 113–116, Tokyo, 1986.

[13] T. Abe and M. Honda. Sinusoidal model based on instantaneous frequency attractors. IEEE Tr. Audio, Speech and Lang. Proc., 14(4):1292–1300, 2006.

[14] Ellis, Daniel P.W., Cotton, Courtenay V. "The 2007 LabRosa Cover Song Detection System," Music Information Retrieval Evaluation Exchange, 2007.

[15] M.-L. Zhang and Z.-H. Zhou. ML-KNN: A Lazy Learning Approach to Multi-Label Learning. Pattern Recogn., 40(7):2038-2048, 2007.

[16] X. Hao, X. Tao, C. Zhang, and Y. Hu. An Effective Method to Improve kNN Text Classifier. In Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, volume 1, pages 379-384, 2007.

[17] K. Z. Mao, K. C. Tan, and W. Ser. Probabilistic Neural- Network Structure Determination for Pattern Classification. IEEE Transactions on Neural Networks, 11:1009-1016, 2000.

[18] P. K. Patra, M. Nayak, S. K. Nayak, and N. K. Gobbak. Probabilistic Neural Network for Pattern Classification. IEEE Proceedings of the 2002 International Joint Conference on Neural Networks, vol. 2:1200 - 1205, 2002.

[19] V.N.Vapnik, "The Nature of Statistical Learning Theory", Springer-Verlag, New York, 1995.

[20] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition", Data Mining and Knowledge Discovery, 2 (2), pp. 121–167, 1998.

[21] *Schatzman, James C,* "Accuracy of the discrete Fourier transform and the fast Fourier transform". SIAM Journal on Scientific Computing. *17: 1150–1166, 1996.*

[22] Mei Jiansheng, Li Sukang. Tan Xiaomei," A Digital Watermarking Algorithm Based On DCT and DWT", Proceedings of the 2009 International Symposium on Web Information Systems and Applications (WISA'09) Nanchang, P. R. China, May 22-24, 2009, pp. 104-107.

[23] An Overview of Non-contact Photoplethysmography, Peck Y S Cheang and Peter R Smith, ELECTRONIC SYSTEMS AND CONTROL DIVISION RESEARCH, 2003.

[24] Frigo, M. and S. G. Johnson, "FFTW: An Adaptive Software Architecture for the FFT,"Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Vol. 3, 1998, pp. 1381-1384.

[25] Zanaty EA, Afifi A, Khateeb RE. Improving the accuracy of support vector machines via a new kernel functions. Int J IntellComputSci 2009;1:55–67.

[26] Zanaty EA, Aljahdali S, Cripps RJ. Accurate support vector machines for data classification. Int J Rapid Manuf 2009;1(2).

[27] EktefaMohammadreza, Sidi Fatimah, Ibrahim Hamidah, JabarMarzanah, Memar Sara. A comparative study in classification techniques for unsupervised record linkage model. J ComputSci 2011;7:341–7.

[28] SonkambleBalwant A, Doye D. A novel linear-polynomial kernel to construct support vector machines for speech recognition. J ComputSci 2011;7:991–6.

[29] Zanaty EA, Afifi A. Support vector machine (SVMs) with universal kernel. ApplArtif. Intell. 2011;25(7):575–89.

[30] DeCoste D, Mazzoni D. Fast query optimized kernel machine classification via incremental approximate nearest support vec-tors. In: Proceedings of the 20th international conference on machine learning, Washington, ICML 2003; 2003. p. 115–22.

[31] Scho¨lkopf B, Smola AJ. Learning with kernels. Cambridge, MA: The MIT Press; 2002.

[32] Scho¨lkopf BS, Mika CJ, Burges P, Knirsch RR, Muller KG. Input space versus feature space in kernel-based methods. IEEE Trans Neural Network 1999;10(5):1000–17.

[33] Burges CJ, Scho¨lkopf B. Improving the accuracy and speed of support vector machines. Neural Inform Proc Syst 1997;9:375–81.

[34] Burges CJ. A tutorial on support vector machines for pattern recognition. Data Min Knowledge Discovery 1998;2(2):121–67.

[35] Scho¨lkopf B, Burges C, Smola A. Advances in kernel methods support vector learning. MIT Press; 1998.

[36] Taylor SJ, Bartlett PL, Willianmson RC, Anthony M. Structural risk minimization over data-dependent hierarchies. IEEE Trans Inf Theory 1998;5:1926–40.

[37] Making JT. Large-scale SVM learning practical. In: Advances in kernel methods-support vector learning. Cambridge: MIT Press; 1999. p. 42–56.

[38] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, Choosing multiple parameters for support vector machines, Machine Learning 46 (1–3) (2002) 131–159.

[39] Suresh Yerva, Smita Nair and Krishnan Kutty,‖ Lossless Image Compression based on Data Folding,‖IEEE, pp. 999-1004, 2011.

[40] T. Tsai, T. Prätzlich, M. Muller, "Known-artist live song id: a hashprint approach", *International Society for Music Information Retrieval Conference*, 2016.

[41] PremSeetharaman, FatemehPishdadian, Bryan Pardo, "Music/Voice separation using the 2D fourier transform", Applications of Signal Processing to Audio and Acoustics (WASPAA) 2017 IEEE Workshop on, pp. 36-40, 2017.