: For Accurate Protein Structural Class Prediction

First Author,[1,*] Second Author,[2] and Third Author[3]
[1]First-Third Department, First-Third University
Address Including Country Name
[2]Second Department, Second University
Address Including Country Name
*first.author@first.com

Abstract—Proteins are the building blocks of all cells in both human and all our living creatures of the world. Most of the work in the living organism is performed by Proteins. Proteins are polymers of amino acid monomers which are biomolecules or macromolecules. The tertiary structure of protein represents the three-dimensional shape of a protein. The functions, classification and binding sites are governed by protein's tertiary structure. If two protein structures are alike then the two proteins can be of the same kind. To detect the similarity of proteins accurately in real time is crucial in the research. In this paper, we present an analysis on local binary pattern histogram,Wavelet transformed Local Binary Pattern Histogram, Separate Row Multiplication Matrix with Uniform Local Binary Pattern Histogram, Neighbour Block Subtraction Matrix with Uniform Local Binary Pattern Histogram and Atom Bond for protein structural class prediction. We have used them on the distance matrix of α carbons of proteins which are used as an image for feature extraction. The experiments were done on a 40 percent reduced dataset of pbd files. We have demonstrated the usefulness of this feature over a large variety of supervised machine learning algorithms. We propose the use of Random Forest as the best performing classifier on this dataset using the selected features.

Index Terms—Protein Structural Similarity , Supervised Learning , Wavelet transform , Local Binary Pattern , Histogram

I. Introduction

Protein tertiary structure comparison is very important in many applications of modern structural biology, drug design, drug discovery, in studies of protein-protein interactions and other fields. This is especially significant because the structure of a protein is more protected than the protein sequence [1]. Many works have been done to find protein binding [2].

Comparison of protein structure has been done in many works of literature by alignment of distance matrices [3], using iterated double dynamic programming [4], using elastic shape analysis[5] and many other techniques. The most common way of comparing protein tertiary structure is to treat the protein as a three-dimensional object and superimpose one on another. Different distances are used to calculate the differences between the proteins.

The distance matrix of α carbon can be seen extensively used in [6] [7] as a feature which represents the tertiary structure of a protein chain. This feature is used as a feature vector which represents the structure of a protein to measure either similarity or dissimilarity to measure and compare the feature vectors with one another in pattern recognition literature. A mapped two-dimensional feature matrix is created from the 3D coordinate data of protein. The intra-molecular distance is used to make the α carbon distance matrix which mirrors the tertiary structure of a protein and the conserved elements of the secondary structure in it. With an input matrix size of N x N, the distance matrix based exact algorithms run in 0(N!) time [8].

An image is basically a matrix of N x N dimension with corresponding data in each cell. Thus the distance matrix can be used as an image. Basically, three types of features can be generated from an image: pixel based, filter based and computationally generated features. Pixel-based features e.g histograms are simplistic and dependent on the capability of classification algorithms. Filter based methodologies transform the original image to use feature extraction methods. Refined algorithms are used to segment and other various algorithms are used to detect different features.

Using ideas from computer vision and utilizing it in protein structure retrieval is not uncommon in the field. ProteinDBS server [9] implement a similar approach in [10] by Chi et al. Texture features from the original size images and diagonally partitioned images were extracted by Chi et al. CoMOGrad and PHOG [8] also used images to extract their two novel feature whereas we are extracting histograms of local binary pattern images from the original image.

In this paper, we propose the combination of local binary pattern histogram,Wavelet transformed Local Binary Pattern Histogram, Separate Row Multiplication Matrix with Uniform Local Binary Pattern Histogram, Neighbour Block Subtraction Matrix with Uniform Local Binary Pattern Histogram and Atom Bond features to be used for protein similarity measurement. We extract the distance matrix of α carbon of a protein from PDB file and use the distance matrix as an image to extract our first four features and Atom Bond is extracted from the PDB files. We have used a large variety of classification algorithms to test the extracted features. We are also going to show the results and comparative study of different implementation methodologies such as wavelet and pyramid histogram based features [11] and CoMOGrad and PHOG. The method we have proposed is able to produce a better result on some classification algorithm over the previous methods on the same benchmark.

Rest of the paper is organized as following: Section II briefly presents a literature review of the related work; Section III describes the methodology and materials proposed in this paper; experimental results are shown in Section IV with a discussion and the paper conclude in

Section V.

II. Related Works

There are experiments performed to compare protein structure as three-dimensional objects. Score function based on different distance metrics to find similarity and dissimilarity as a measurement has been proposed by these methods. The most prominent improvement of the literature is presented briefly below.

A. DALI

Distance Alignment Matrix Method(DALI)[6] calculates an alignment score by finding an absolute alignment between the α carbon distance matrices of proteins. A distance matrix is created by breaking the input structure into hexapeptide fragments and evaluating the contact patterns(pair-wise) between them and making a list with a matching score by saving the matching pairs[3]. The final matching score and overall alignment are made by gathering pairs in the correct order. Monte Carlo optimization is used for the assembling.

B. CE

Combinatorial Extension (CE)[12] is comparable to DALI because it creates a series of fragments by breaking each structure in the query set and later attempts to reassemble toward a complete alignment. Protein structures are compared by using combinatorial extension and Monte Carlo optimization. The computational cost is quite huge to implement this method despite having good accuracy. Thus a real-time web service cannot be implemented due to its cost ineffectiveness.

C. SSAP

The Sequential Structure Alignment Program (SSAP)[13] uses β carbons unlike the other methods using α carbon of protein in structural alignment except for glycine. Double Dynamic programming is used to produce atom-to-atom vectors which is based on structural alignment in structure space. Inter-residue distance vectors amid every individual residue and its nearest non-contiguous neighbors on each protein are first generated. The vector differences amid neighbors are created in a series of matrices. Optimal local alignments are found from each resulting matrix by applying dynamic programming. A 'summary' matrix is created from the summed up optimal local alignment. A comprehensive structural alignment is resolved by applying dynamic programming again.

D. FATCAT

Flexible structure AlignmenT by Chaining Aligned fragment pairs with Twists (FATCAT)[14] treats the protein structure like a fixed body. It produces good results for maximum cases with other fixed body approaches[15]

E. ProteinDBS

ProteinDBS[9] compares α carbon distance matrix images by using some common features of CBIR(Content Based Image Retrieval). It correlates only some particular image features thus it performs much faster than the previous ones. The drawback of ProteinDBS is the expensive cost of computation.

F. TM-align and SP-align

The most well known method for protein structure alignment is TM-align[16] [17].Finding an optimal alignment and an alignment matrix it computes the TM-Score.

SP-ALign [18] is also a popular approach which is similar to TM-align. The difference between the two lies in the alignment algorithm and the alignment score.

G. CoMOGrad and Phog

CoMOGrad stands for Co-occurrence Matrix of the Oriented Gradient of Distance Matrices and PHOG stands for Pyramid Histogram of Oriented Gradient [8]. This methodology also uses the α carbon distance matrix of protein. The dimension of all distance matrix is converted to 128 × 128.

In CoMOGrad, the gradient angle and magnitude is computed from the distance matrix and the values are quantized. Quantization is a compressing technique which compresses a range of values to a single quantum value. In this methodology, the values are quantized to 16 bins which produce a co-occurrence matrix which is 16 × 16 matrix. The matrix is converted into a vector of size 256.

Quadtree from the distance matrix is created with the desired level in PHOG. Gradient Oriented Histogram of each node is calculated with the preferred number of bins and bin size. In gradient oriented histogram an image is divided into small sub-images called cells and histogram of edge orientations are accumulated within the cell. The combined histogram entries are used as the feature vector describing the object. Total features which are the multiplication of total nodes and number of bins are incorporated in the vector with the size of the total number of features. The vector is normalized by dividing it with the sum of its components.

III. Our Method

In this section, we are going to describe our methodology. Images are created from the α carbon of protein collected from the PDB files of the given dataset.BMSULBP-Hist ,LBP histogram and Wavelet Transformed LBP histogram features are extracted from each images referring to total seven classes of protein. Synthetic Minority Oversampling Technique(SMOTE) is used to remove class imbalance problem. K-fold cross-validation with three fold was used to test the capability and efficiency of the dataset. The block diagram of the methodology used in this paper is given in Figure 1.

A. Dataset

We have used 40 percent ID filtered subset of PDB-style files for SCOPe domains version 2.03 [19] as our dataset. It contains a total of 12119 PDB files. Each PDB files contains SCOP(e) concise classification string (sccs) which respectively describes class, fold, superfamily, and family. In this literature, we are going to experiment only with the class of protein. In the dataset, there are total seven protein classes. The names of the protein classes can be found in Table II.

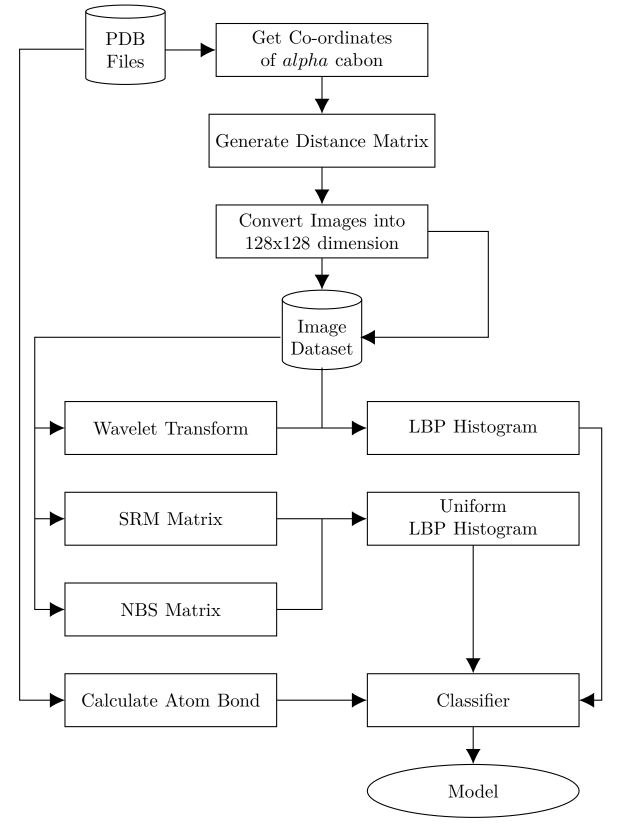

Fig. 1. Block diagram of the methodology used in our paper

B. Image Generation

We have generated Images of Protein Structure according to the methodology described in CoMOGrad and PHOG [8]. The number of α carbons of protein can be found in the PDB file of the protein. The total number of α carbon atoms are calculated from the PDB file and the x,y and z coordinates of the α carbon stored. They are used to generate a distance matrix. The matrix is used as the image of the protein structure of that particular protein. The generated images are black and white in nature.

C. Scaling Images to Same Dimension

The dimension of protein images is based on the total number of α carbon they have. So, every individual protein images are different from the other. Therefore, the images should be scaled to the same dimension. CoMOGrad and PHOG have used Bi-cubic interpolation and wavelet transform to scale all the protein images into 128 x 128 dimension[8]. During the Bi-cubic interpolation step, most of the images were in 128x128 dimension so in the wavelet transform step they scaled all the images to that dimension. Thus, we have directly scaled the images to 128x128 dimension. We have used both real and scaled images to examine the results.

D. Feature Extraction

Our first four feature groups are types of histograms and the fifth feature group is about the prognosis of the atoms. The histograms were made from both scaled and unscaled images.

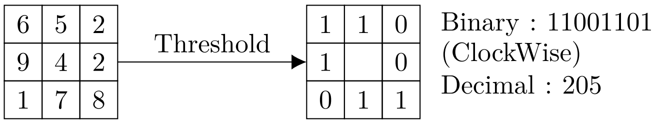

Fig. 2. An example of basic LBP

1) Local Binary Pattern Histogram: The work of Ojala et al.[20] popularized LBP. Although it was first narrated in 1994[21].Local Binary Pattern computes the local representation of the texture of an image as a texture descriptor. Comparing each pixel with its neighboring pixels the local representation is created. The image is transformed into a grayscale image. In a 3x3 neighborhood, the center pixel value is calculated by comparing with its eight neighboring pixels. Each comparison gives a result of either 0 if the center pixel value is greater then the comparing neighbor pixel or 1 for the latter. A clockwise direction starting from the top-left one provides a binary number. The binary number is converted to a decimal number and the value is placed in the center pixel. LBP codes or Local Binary Patterns are the obtained binary numbers. An example of a basic Local Binary Pattern is given in Figure 2.After calculating the value for each pixel of the image, a histogram is calculated. A 3 x 3 neighborhood has $2^8 = 256$ possible patterns, thus the values range from 0 to maximum 255 in each pixel of the image. The total number of bins of the histogram is thus 256. We would get 256 attributes from each image.

There exist many variations of LBP. Other then the basic LBP , we have used Uniform Local Binary Pattern[20].

2) Wavelet transformed Local Binary Pattern Histogram(WtLBP-Hist): We have used Haar wavelet transform[22] for our wavelet transformation. It is based on lifting scheme. Wim Sweldens developed by Lifting scheme[23]. The image is converted into three two dimensional matrices for storing blue, green and red value of each pixel. The rows and columns of the three matrices and protein image are equal. Haar wavelet transformation is applied on the three matrices and the corresponding pixel value of the three images are added to generate the wavelet image. LBP techniques are then used on wavelet image to get 256 attributes.

3) Separate Row Multiplication Matrix with Uniform Local Binary Pattern Histogram(SRM Matrix-ULBP-Hist): The image is split into 3x3 matrices. From each matrix, we get 3 rows with the dimension of 1x3. By multiplying each row with the same 3x3 matrix, we get three result matrix consisting of 1x3 dimension. Each cell is divided by 100. The results are then put in the 3x3 matrix in accordance with the row numbers. The color intensity of an image is between 0 to 255. So, if the value of any cell of the result matrix is greater than 255, then the value is replaced with 255. After applying this technique, the uniform local binary pattern is applied. From Figure 3, (a) presents a 3x3 section of matrix and the rows, (b) exhibits the result of multiplication, (c) shows the value after dividing by 100, (d) shows the replacement result of value greater than 255 and (e) shows a 3x3 matix section after SRM-Matrix transformation.

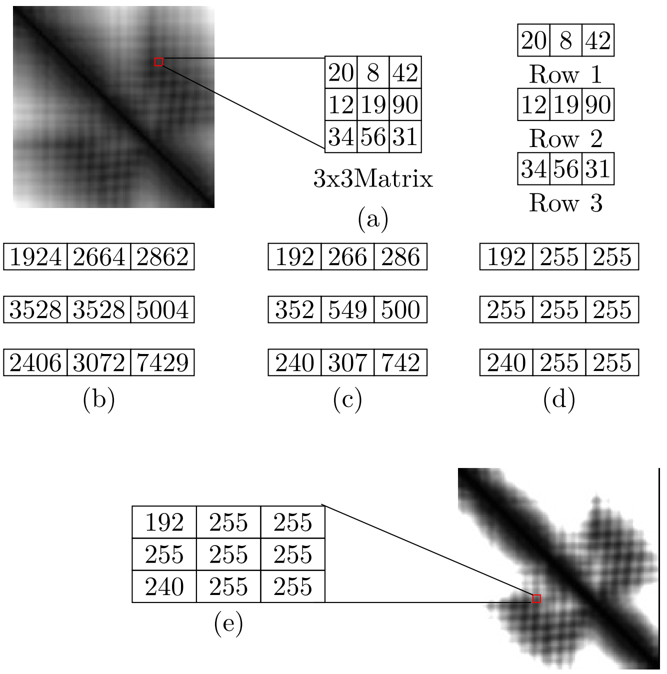

Fig. 3. An example of Separate Row Multiplication Matrix with Uniform Local Binary Pattern Histogram

Another variation of the LBP is called uniform pattern[20]. Some binary patterns occur more generally in texture images. If the binary pattern comprises of at most two 0-1 or 1-0 transitions when the bit pattern is held circular then the pattern is called uniform. For instance, 01000000 has 2 transitions, 00000111 has 2 transitions which are uniform pattern on the other hand 01010100 has 6 transitions,11001001 has 4 transitions which are not uniform. A neighborhood with the dimension of 3x3 has $2^8 = 256$ possible patterns with 58 of them being uniform. For estimating the histogram, every uniform pattern gets a separate bin while a single bin is allotted for all non-uniform patterns. Therefore, from a uniform binary pattern, we get the histogram of total bin size of 59.

4) Neighbour Block Subtraction Matrix with Uniform Local Binary Pattern Histogram(NBS Matrix-ULBP-Hist): Blocks are of the same dimension, 3x3. Two blocks of matrices are considered neighbors for this method if the center cells are neighboring. Because of this, the value of the last two columns of the first block and first two columns of the second block are same. The two blocks of matrices are subtracted and the result is set in the place of the first block. If any of the cells have any negative number, then 0 is placed instead of the negative value. The replacing of value is made because the histogram bin begins from zero. Uniform local binary pattern is then used to compute the histogram.

5) Atom Bond Features: First of all, we've identified unique atoms amidst all the protein PDB files. From each protein PDB file, we've counted occurrences of each atom. Then we've taken the percentage as features of each atom among all the atoms that each protein has. Then we've taken first 100 sequential atoms and used their atomic mass as the feature. Then we've counted the bond that each pair of atoms has in a particular protein using atomic distance based on a threshold value. Finally, we've taken the percentage as the feature of the bond of each unique pair of atoms among all the bonds that the protein has.

TABLE I
Feature Groups

Identifier	Feature Group Name	Number of Features
A	LBP-Hist	256
B	WtLBP-Hist	256
C	SRM Matrix-ULBP-Hist	59
D	NBS Matrix-ULBP-Hist	59
E	Atom Bond	116

TABLE II
Protein Classes and its Corresponding Instances

Class Name	Total Instances
Small Proteins	640
All α Proteins	2195
α and β proteins(a/b)	3305
α and β proteins(a+b)	3006
Membrane and cell surface proteins and peptides	204
All β proteins	1485
Multi-domain proteins(α and β)	219

Summary of all the feature groups used in this paper is given in Table I.

E. Re-evaluate Dataset

To perform benchmark analysis, we received the dataset generated from the CoMOGrad and PHOG literature [8]. As the SCOPe-sid is unique for every variant of the protein, we have created the dataset based on the proteins which are both on the 40 percent Id filtered subset and CoMOGrad and PHOG paper. After analyzing we have found that there are total 11052 instances in both of our feature groups and CoMOGrad and PHOG features. The seven classes and the total number of instances of each class are given in Table II.

F. Removing Multiclass Imbalance Problem

From Table II it can be noted that the classes are imbalanced. To balance the classes, we have used Synthetic Minority Over-sampling Technique (SMOTE)[24].

In Weka, the percentage of SMOTE indicates that how many more instances would be generated. As the highest number of instance a class has is 3305, we have oversampled our instances close to that number. If x is denoted by the highest number of instance among all the classes and y denoted by a class which we will SMOTE then the equation for the percentage calculation is shown in (1).

$$\frac{x-y}{y} * 100 \qquad (1)$$

We have used 5 nearest neighbors to generate the oversampled instances. After applying SMOTE to all data sets, The total number of instances of each dataset close to 23132.

G. Classifier

We have used five classifiers: K-Nearest Neighbor (KNN), Naive Bayesian Classifier, Support Vector Machines (SVM), Adaptive Boosting (AdaBoost) and Random Forest. A concise description of the classifiers is given in this section.

1) K-Nearest Neighbour (KNN): K-nearest neighbour algorithm (KNN) [25] is a similarity-based classification technique. It is a lazy classification technique. Distance metrics are used for each instance of the whole dataset for calculating the K nearest neighbors. The labels of the nearest neighbors decide the label of the test instances. It works poorly for high dimensional data. Euclidean distance, Hamming distance, Manhattan distance, Minkowski distance, Tanimoto distance and Jaccard distance are used for similarity measures.

2) Naive Bayesian Classifier: Naive Bayesian classifier [25] is based on probabilistic inference of samples observed where the decision variable and the features form a very naive structure of Bayesian Network. Naive Bayesian classifiers work best for image recognition and text mining.

3) Support Vector Machine (SVM): Support Vector Machine[25] works by creating and separating hyperplane for a given dataset by sampling different classes which are separated by maximum width.

4) Adaptive Boosting (AdaBoost): Adaptive Boosting classifier [25] is a meta-classifier which aims to make a strong classifier using a set of weak classifiers. The classifiers whose performance are marginally better than random classifiers are called weak classifiers.

5) Random Forest: Random Forest [25] is an ensemble classifier.A decision tree is created in each iteration with features taken randomly. It samples selected features using bootstrap aggregating.

H. Performance Evaluation

Separate independent test set or cross fold sampling method is used by researchers for performance evaluation. They are used to check the stability of the model. As k-fold cross-validation overcomes the problem of over-fitting it is preferred by researchers for performance approximation. We have used k-fold cross-validation technique.K-fold cross validation splits the data into k partitions and then use each partition as a test set with each iteration where the training data is the rest of the data. We have used accuracy as the performance metric in this paper. The percentage of correctly classified instances to the total number of instances is termed as accuracy.

IV. Experiments Results

In this section, we are going to describe the experiments conducted in this study. Some of the experiments were carried out in a Dell Inspiron 15 Laptop Computer of 3000 series with 4 GB RAM and 240 GB SSD hard drive, some of them were carried out in Intel Core i5 4590 processor personal computer with 12 GB Ram and 2048MB ATI AMD Radeon R7 200 Series Graphics Card and others were experimented in <SERVER SPECIFICATION>. Java language is used to implement all the programs using the Eclipse IDE and Java 8 standard edition. Features were generated using The OpenCV software library [26]. Weka tool [27] was used to implement the classification algorithms used in this paper.

TABLE III
Accuracy of different classifiers of protein images

Image Type	Feature Type	Classifiers				
		KNN	Naive Baysian	SVM	Ada Boost	Random Forest
Non Scaled	A	68.69	33.15	65.62	83.37	87.50
Non Scaled	B	77.28	32.17	67.24	83.52	86.88
Scaled	A	74.06	53.06	78.58	83.30	85.22
Scaled	B	78.69	49.79	79.80	84.68	86.68
Scaled	C	84.10	51.02	71.02	83.20	85.11
Scaled	D	81.58	56.40	72.25	81.89	83.99
	E	66.96	21.79	44.49	62.26	69.92
Scaled + Non Scaled	AB	78.62	41.87	83.08	83.08	86.08
Scaled	CD	84.65	57.55	78.10	84.24	86.28
Scaled + Non Scaled	ACD	84.51	41.96	81.99	86.97	88.66
Scaled	BCD	81.87	55.02	84.29	85.64	87.37
Scaled + Non Sclaed	ABCD	84.38	36.90	83.53	86.48	88.77
Scaled + Non Sclaed	ABCDE	76.47	36.86	85.78	87.57	89.03

A. Parameters used for the classifiers

A different set of parameters were used for each classifier. A linear searching was used with no distance weighting for KNN. In case of the Naive Bayesian Classifier, SVM, a polynomial kernel was used with $c = 1.0$ and $\epsilon = 1.0w^{-2}$. Data was normalized before supplying to the classifier. J48 decision tree classifier was used in Adaboost classifier as the weak base classifier. Classifier number of iterations was set to 100 for Random Forest.

B. Analysis of Features

In this section, we are we are going to present the analysis of our features. Results in terms of average accuracy in 3-fold cross-validation of protein images are given in Table III. The highest percentage of correctly classified instances achieved for each of the classifiers are indicated by the boldly faced values of the table.

After running the experiments for our five feature groups ABCDE classifies the highest percentage of correct instances in Random Forest, Adaboost and SVM among all other feature groups. Feature group CD provides the highest accuracy in KNN and Naive Bayesian. As the whole combination of all feature groups accuracy gives the highest percentage than any other feature group, thus we conclude that the best performing feature group combination is ABCDE and the best classifier is Random Forest classifier.

C. Comparison with other methods

In this section, we compare the performance of our proposed method with CoMOGrad and PHOG[8] along with our previous published literature Wavelet and Pyramid Histogram Features for Image Based Leaf Detection[11]. For comparison with our methodology in this literature,

TABLE IV
Comparison of the method proposed features in this paper with [8] and [11]

Feature Type	Classifiers				
	KNN	Naive Baysian	SVM	Ada Boost	Random Forest
Karim et al.[8]	87.41	59.50	87.67	84.19	85.49
Ahmed et al.[11]	69.36	36.22	67.30	79.92	84.58
this paper	84.65	57.55	85.78	87.57	89.03

we applied CoMOGrad and Phog techniques and Wavelet and Pyramid Histogram techniques in our dataset of 11052 instances and later applied SMOTE for reducing class imbalance problem. We conducted experiments with different classifiers using the same parameters as we did for feature analysis with the feature groups. The results are given in Table IV. From Table IV it can be comprehended that our feature group ABCDE outperforms CoMOGrad and PHOG in Random Forest and in Adaboost. CoMOGrad and PHOG surpassed our feature groups in KNN, Naive Bayesian and SVM.It can be noted that the combination of our feature groups are three-fourths of CoMOGrad and PHOG. It also can be discerned that the accuracy percentage in Random Forest is higher than all the classifier results. Thus, our novel features can classify more instances than CoMOGrad and PHOG. We have also noticed that our feature groups outperform the features of our previous literature[11] on all classifiers.

D. Discussion

We have revealed the precedence of our methodology over CoMOGrad and PHOG [8] and Wavelet and Pyramid Histogram Features for Image Based Leaf Detection[11]. The same feature groups were used for leaf detection [11] with the dataset consisting of RGB images of leaves. Unlike only gray histogram used on this paper, blue, green and red histograms were used to generate features in each feature group and the accuracy result of each classifier was high. The distance matrix of α carbons or the protein images were black and white, thus only gray histogram was used as a feature.

We also used Scale-invariant feature transform(SIFT)[28] methodologies in our experiments. Each descriptor has a 128-dimensional feature vector. The number of the descriptors of SIFT from every image is not specific so we cannot use traditional machine learning techniques. Hence to apply traditional machine learning procedure and specify the feature vector, we split the image into 16 slices and took one descriptor from each of the slice images. Therefore we got 2048 number of attributes(8x16) from each image. We tested the dataset with the same classifiers mentioned in this paper. The results didn't turn up to be better or close to our proposed methodology in this literature.

V. Conclusions

In this paper, we showed how accurately we can detect protein classes using the combination of our feature group ABCD of protein images. As the advancement of deep learning, neural network, and many other deep learning techniques are being used to classify images, many remarkably interesting applications can be made. For our future advancement, we wish to introduce new features to improve accuracy, use new tools and explore other fields of computer vision such as human emotion detection.

Acknowledgment

We thank Rezaul Karim for sharing the SQL dataset files and algorithms to generate Distance Matrix from PDB files for CoMOGrad and PHOG and Md Mosharof Hossain for lending his personal computer.

References

[1] C. Chothia and A. M. Lesk, "The relation between the divergence of sequence and structure in proteins." The EMBO journal, vol. 5, no. 4, pp. 823–826, 1986.

[2] G. P. Brady and P. F. Stouten, "Fast prediction and visualization of protein binding pockets with pass," Journal of computer-aided molecular design, vol. 14, no. 4, pp. 383–401, 2000.

[3] L. Holm and C. Sander, "Protein structure comparison by alignment of distance matrices," Journal of molecular biology, vol. 233, no. 1, pp. 123–138, 1993.

[4] W. R. TAYLOR, "Protein structure comparison using iterated double dynamic programming," Protein Science, vol. 8, no. 3, p. 654–665, 1999.

[5] S. Srivastava, S. B. Lal, D. Mishra, U. Angadi, K. Chaturvedi, S. N. Rai, and A. Rai, "An efficient algorithm for protein structure comparison using elastic shape analysis," Algorithms for Molecular Biology, vol. 11, no. 1, p. 27, 2016.

[6] L. Holm and C. Sander, "Dali/fssp classification of three-dimensional protein folds," Nucleic acids research, vol. 25, no. 1, pp. 231–234, 1997.

[7] A. P. Singh and D. L. Brutlag, "Hierarchical protein structure superposition using both secondary structure and atomic representations." in Ismb, vol. 5, 1997, pp. 284–293.

[8] R. Karim, M. M. A. Aziz, S. Shatabda, M. S. Rahman, M. A. K. Mia, F. Zaman, and S. Rakin, "Comograd and phog: From computer vision to fast and accurate protein tertiary structure retrieval," Scientific Reports, vol. 5, pp. 13 275 EP –, Aug 2015, article. [Online]. Available: http://dx.doi.org/10.1038/srep13275

[9] C.-R. Shyu, P.-H. Chi, G. Scott, and D. Xu, "Proteindbs: a real-time retrieval system for protein structure comparison," Nucleic Acids Research, vol. 32, no. suppl_2, pp. W572–W575, 2004.

[10] P.-H. Chi, G. Scott, and C.-R. Shyu, "A fast protein structure retrieval system using image-based distance matrices and multidimensional index," International Journal of Software Engineering and Knowledge Engineering, vol. 15, no. 03, pp. 527–545, 2005.

[11] A. A. N. Ahmed, H. M. F. Haque, A. Rahman, M. S. Ashraf, and S. Shatabda, "Wavelet and pyramid histogram features for image-based leaf detection," in Emerging Technologies in Data Mining and Information Security, A. Abraham, P. Dutta, J. K. Mandal, A. Bhattacharya, and S. Dutta, Eds. Singapore: Springer Singapore, 2019, pp. 269–278.

[12] I. N. Shindyalov and P. E. Bourne, "Protein structure alignment by incremental combinatorial extension (ce) of the optimal path." Protein engineering, vol. 11, no. 9, pp. 739–747, 1998.

[13] C. A. Orengo and W. R. Taylor, "[36] ssap: sequential structure alignment program for protein structure comparison," in Methods in enzymology. Elsevier, 1996, vol. 266, pp. 617–635.

[14] Y. Ye and A. Godzik, "Flexible structure alignment by chaining aligned fragment pairs allowing twists," Bioinformatics, vol. 19, no. suppl_2, pp. ii246–ii255, 2003.

[15] M. Shatsky, R. Nussinov, and H. J. Wolfson, "Flexible protein alignment and hinge detection," Proteins: Structure, Function, and Bioinformatics, vol. 48, no. 2, pp. 242–256, 2002.

[16] Y. Zhang and J. Skolnick, "Tm-align: a protein structure alignment algorithm based on the tm-score," Nucleic acids research, vol. 33, no. 7, pp. 2302–2309, 2005.

[17] L. Zhang, J. Bailey, A. S. Konagurthu, and K. Ramamohanarao, "A fast indexing approach for protein structure comparison," BMC bioinformatics, vol. 11, no. 1, p. S46, 2010.

[18] Y. Yang, J. Zhan, H. Zhao, and Y. Zhou, "A new size-independent score for pairwise protein structure alignment and its application to structure classification and nucleic-acid binding prediction," Proteins: Structure, Function, and Bioinformatics, vol. 80, no. 8, pp. 2080–2088, 2012.

[19] N. K. Fox, S. E. Brenner, and J.-M. Chandonia, “Scope: Structural classification of proteins—extended, integrating scop and astral data and classification of new structures,” Nucleic acids research, vol. 42, no. D1, pp. D304–D309, 2013.

[20] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Transactions on pattern analysis and machine intelligence, vol. 24, no. 7, pp. 971–987, 2002.

[21] T. Ojala, M. Pietikainen, and D. Harwood, “Performance evaluation of texture measures with classification based on kullback discrimination of distributions,” in Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on, vol. 1. IEEE, 1994, pp. 582–585.

[22] R. S. Stanković and B. J. Falkowski, “The haar wavelet transform: its status and achievements,” Computers & Electrical Engineering, vol. 29, no. 1, pp. 25–44, 2003.

[23] W. Sweldens, “The lifting scheme: A construction of second generation wavelets,” SIAM journal on mathematical analysis, vol. 29, no. 2, pp. 511–546, 1998.

[24] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” Journal of artificial intelligence research, vol. 16, pp. 321–357, 2002.

[25] M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of machine learning. MIT press, 2012.

[26] G. Bradski, “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools, 2000.

[27] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: an update,” ACM SIGKDD explorations newsletter, vol. 11, no. 1, pp. 10–18, 2009.

[28] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004.