CHAPTER 5

Conclusion

We presents a new algorithm named “incorporating uncertainty into string grammar K-nearest neighbor” for identifying or classification data into appropriate groups and compared the performance of the algorithm to the existing research. Next, we produced 7 algorithms were evaluated on 10 public datasets for face recognition consisting of ORL, MIT-CBCL, Georgia Tech, FEI, JAFFE, Pain Expression, Senthilkumar, PICS, Yale, and CMU AMP databases.

Also, we used 7 algorithms on 5 public datasets for facial expression recognition that consisted of JAFFE, YALE, CMU AMP, CK+ (327 VDO), and UNBC - McMaster Shoulder Pain Expression Archive databases.

Furthermore, we used 7 algorithms on 3 datasets that consisted of Kimia-216, Image Hjpg and USPS databases for testing our algorithm on the other side, i.e., the properties and the problems.

We considered all 7 sgFKNNs and found that each algorithm has the same properties but can be separated into 2 groups as:

The first group consists of sgFKNN1, sgFKNN2, sgFKNN3, and sgFKNN7. With a test sample, the algorithms get high accuracy rate than the second group. The second group of the algorithms consists of sgFKNN4, sgFKNN5 and sgFKNN6 whose results require less computing time because they only calculate the distance between the test sample with the prototype. The results of classification from the second group are considered to be worse than the first group’s experiment, but the accuracy still above 10% when using the appopriate number of prototypes.

The face recognition and facial expression experiments showed the higher accuracy rate than most works in 13 datasets. Although some algorithms have higher accuracy rates than the proposed algorithms in some datasets, our algorithm does not require images to be cropped beforehand which differs from most of the existing algorithms.

In addition, it is difficult to find the optimal of *K* values in some datasets. In most of our experiments we found that the optimal values of *K* are 1, 3, 5, 7, and the minimum number of images for each person minus 1 to obtain the highest accuracy rate.

The last problem, we found that the process the part of string generation in our experiment. The best of accuracy rate was found when using 1600 symbols for 1 string sequence which system requires higher computing time. Figure 4.56 shows that the accuracy rate was dropped when using 2500 symbols, whereas 1600 symbols for 1 string produced the best results.

However, most of our experiments found that when we used 100 symbols for 1 string sequence, the accuracy rates were good enough and the system required less computing time.

Although the second group has lower accuracy rates than the first group but it is faster. However, the second group cannot use the *K* value more than the number of prototypes. In most of the first group's experiments, the values of *K* that provide high accuracy rates are 1, 3, 5, 7 and *M*-1 where *M* is minimum number of images for each class. We found that we cannot use *K* equal to *M*-1 for the second group. That is the *K* value can be only 1 and 3 for higher accuracy rates.

However, the values of *K* affect the membership values as follow:

- If *K* = 1, membership values can be only 0 or 1 when using crisp membership initialization.
- If *K* increases, the membership value in each class decreases but also still between 0 and 1.

Moreover, the accuracy rate on the membership values in the crisp initialization greater than in the fuzzy initialization and when K increases the membership values decrease, which is shown in appendix A.2.

We compared the proposed method with other methods that are based on the same datasets as in the above section. The overall results were better than the ones from the other methods. Although our method provided worse results in some datasets, it is because we did not crop images beforehand while most of the existing algorithms did.

However, according to Section 4.2.1, the computational cost of string generation method is around $O\left(N^2\right)$.

The algorithms in the first group which consists of sgFKNN1, sgFKNN2, sgFKNN3, and sgFKNN7 have the same computational time complexities as $O(C \times K \times N \times N \log N) = O\left(N^2 \log N\right)$. For the second group which consists of sgFKNN4, sgFKNN5, and sgFKNN6, the computational time complexities are $O(C \times K \times C \times N \log N) = O\left(N \log N\right)$.

For further works, the USPS dataset is the digit data for which the string generation process should be implemented with the techniques like contour detection. The term contour can be defined as an outline or a boundary of an object such as Freeman Chain Code. Moreover, one possible improvement to the proposed method is to use the algorithm in the data classification on more datasets and improves some variables to achieve more accuracy rates and reduce the time complexity.

The disadvantages of our algorithms are as follows:

- Difficult to find an optimal of K value. This may requires the experimentation by varying K values on the dataset.
- For sgFKNN4 to sgFKNN6, they are difficult to find an optimal of number of prototypes.