# A Simple Resource-Aware Approach to Sketch Recognizers via Style Identification

by

Burak Özen

A Thesis Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Computer Engineering

Koç University

August 26, 2013

Koç University
Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Burak Özen

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Assoc. Prof. Tevfik Metin Sezgin

Assoc. Prof. Engin Erzin

Assoc. Prof. Yücel Yemez

Date:

## ABSTRACT

Sketching is one of the natural mode of communication among humans. With the recent increase in the availability of pen-based devices, a growing trend towards sketch-based interfaces and sketch recognition systems have emerged in Human Computer Interaction. Modern approaches to sketch recognition make heavy use of machine learning technology to maximize recognition accuracies by learning from examples. Although having more training examples is key to the performance of any sketch recognition framework, certain aspects related to the practical use of machine learning technology have surfaced as real issues that need attention. One of these practical issues that hinders the development and deployment of sketch recognition systems is the excessive computational resources. During supervised learning of a sketch recognition system, if a large training dataset is used to train a system model, it costs more training time and results in a bulky model with poor classification performance. In this thesis, we propose a practical, simple, and easy to implement method that sketch recognition practitioners can resort to for partitioning their training data by based on sketching styles of users. Our method leverages the observation that certain groups of people have similar sketching styles, and generating models for smaller groups of people with similar styles reduces training and classification times without a significant sacrifice in recognition accuracies. Our overall system is consisted of two main parts such that in the first part, we partition the all available training data into style sub-groups and in the next part, we designed a system to identify sketching style of an incoming user to assign the user into one of the style groups generated in the first part. We demonstrate the utility of our approach with empirical results obtained from databases of various sizes and characteristics.

# ÖZETÇE

Çizim insanların doğal iletişim araçlarından biridir. Kalem temelli cihazların son zamanlardaki artışı ile birlikte, insan-bilgisayar etkileşimi alanında çizim arayüzleri ve çizim tanıma sistemlerine olan ilgi büyüyen bir eğilim göstermektedir. Çizim tanıma sistemlerine yönelik güncel yaklaşımlar, çizimlerin tanınma oranlarını artırmak adına makine öğrenimi teknolojilerini fazlaca kullanmaktadırlar. Makine öğreniminde sistem var olan örneklerin üzerinden bir öğrenim yapabilmektedir. Her ne kadar bu örneklerin sayısının fazla olması çizim tanıma sisteminin tanıma performansı için önemli bir koşul olsa da, makine öğrenim teknolojilerinin pratik kullanımı hususunda bazı konular dikkat çekmeye başlamıştır. Çizim tanıma sistemlerinin gelişimini aksatan konulardan biri de aşırı veri işleme sorunudur. Bir çizim tanıma sisteminin gözetimli öğrenimi sırasında, eğer sistem büyük bir veri grubu kullanarak eğitilmeye çalışılırsa, bu durum uzun öğrenme zamanına ve sınıflandırma performansı düşük olan hantal bir sistem modeline neden olur. Bu çalışmada biz pratik, basit, uygulaması kolay ve herhangi bir çizim tanıma sistemiyle ilgilenen kişinin eğitim veri setini daha ufak stil gruplarına bölümlendirmesi ve bir kullanıcının çizim stilinin tayin edilmesi için başvurabileceği bir yöntem amaçlanmaktadır. Bizim yöntemimiz belirli insanların çizim stillerinin birbirine benzediği gözleminden faydalanarak, stilleri birbirine benzeyen küçük insan toplulukları için modeller üretip bu sayede bir taraftan öğrenme ve sınıflandırma zamanlarında azalmaya yol açarken, diğer taraftan çizim tanıma oranlarında önemli bir düşüş yaşamamayı amaçlamaktadır. Genel olarak sistemimiz iki temel kısımdan oluşmaktadr. İlk kısımda, insanların çizim stil farklılıklarından faydalanılarak var olan tüm eğitim veri setinin daha küçük stil gruplarına bölümlenmesi amaçlanırken, ikinci kısımda, sisteme yeni gelen bir kullanıcının stilinin belirlenip ilk kısımda üretilen stil gruplarından hangisine ait olduğunun tayin edilmesi amaçlanmıştır.

Yaklaşımlarımızın kullanışlı olduğunu farklı karakterdeki ve boyuttaki çizim veri tabanlarından aldıgımız sonuçlarla göstermiş olduk.

## ACKNOWLEDGMENTS

Foremost, I would like to express my sincere gratitude to my advisor Assist. Prof. Tevfik Metin Sezgin for the continuous support of my M.Sc study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I am really glad to be a part of the Intelligent User Interface Laboratory during M.Sc study. I would like to thank my co-workers : Atakan Arasan, Ayse Küçükyılmaz, Banuçiçek Gürcüoglu, Erelcan Yanık, Çağla Çığ, Özem Kalay and Sinan Tümen. It was fun and inspiring to work with them throughout the two years.

I also want to thank my house-mates : Hüseyin Güler and Seyhan Uçar for their strong support. We shared our house during two years and had a lot of fun together. They provided emotional support at very distressing times for me and convinced me not to give up.

Last but not least, I especially would like to thank my mother Neslihan Özen, my father Şevket Özen and my sister Hatice Özkul. They have been a great and supportive family throughout my life.

I gratefully acknowledge the financial support from the TÜBİTAK throughout my M.Sc study.

# TABLE OF CONTENTS

## LIST OF FIGURES

# NOMENCLATURE

| | |
|---|---|
| $AccuracyRate$ | Recognition ratio of new samples in a trained sketch recognition system |
| $SVM$ | Support Vector Machines |
| $TrainingData$ | Used data that machine learning algorithm learns from |
| $TrainingTime$ | The time which is spent for training session of machine learning algorithm |
| $ClassificationTime$ | The time which is spent for recognizing of new samples |
| $S_{ij}$ | Accuracy of a classifier trained with data from user $i$ on the data of user $j$ |
| $P_{ij}$ | Probability of assigning a sample sketch from object class i to object class j |

Chapter 1

# INTRODUCTION

## 1.1 What is Sketch Recognition?

People naturally use multiple modalities such as speech, handwriting, gesture or sketching while they are interacting with each other. Human Computer Interaction (HCI) concerns the issues that enable natural interaction between users and computers. Therefore, intelligent recognition systems such as handwriting recognition tools, gesture recognition systems or sketch recognition frameworks have attracted great attention recently in this field. Especially, sketch recognition has become one of the most striking fields of study in HCI. As any intelligent recognition systems do, sketch recognition first requires a training process by learning from sketch examples via machine learning technologies. After that, the system can identify new hand-drawn sketches and assigns a class label automatically.

Sketch recognition has been used in various different fields. Military command recognition system is one of the most famous one among them. In army, different commands can be coded as different shapes and due to the intelligent sketch recognition system, commands can be easily recognized and labelled. Moreover, sketch recognition has also become popular in game industry. Lately, some strategy games that are controlled by using sketch commands has emerged. Therefore, we can say that sketch recognition is a hot topic in many fields but still needs attention to develop.

## 1.2 What is the problem with Sketch Recognition?

The increasing availability of pen-based devices such as smart phones, and tablet PCs has lead to substantial interest in building systems capable of interpreting handwriting and hand-drawn sketches. However, despite high recognition rates enjoyed by handwriting recognition systems, sketch recognition systems still lag behind in terms of accuracy.

Handwriting recognition systems owe their high recognition accuracies partly to combining the state of the art in machine learning with large amounts of training data. For example, the MNIST handwritten digit database contains 50000 example patterns collected from 250 writers, which when used to train SVM-based classifiers yield accuracies around 98% [Yann and Corinna, 2009].

Availability of sufficient labeled data is key to the performance of any learning algorithm. Unfortunately, existing approaches to sketch recognition do not scale well with respect to size of the training data. This has lead to efforts that try to reduce time and memory requirements for training and classification through carefully designed training and recognition architectures [Tirkaz et al., 2012], [Zhengxing et al., 2004] or by machine learning techniques such as active learning [Settles, 2009]. However, these methods are quite elaborate, hard to understand, and they require substantial implementation effort.

## 1.3 Thesis Statement

In this thesis, we propose a practical, simple, and easy to implement method that sketch recognition practitioners can resort to for training recognizers in limited time. We leverage the observation that certain groups of people have similar drawing styles, and training models for smaller groups of people with similar styles reduces training times without a significant loss in recognition accuracies. In addition to the advance in training time, resulted smaller style models also reduce classification times required for predicting new coming samples.

Specifically, we show that our method can effectively produce faster models and reduce training times of these models by identifying users with similar sketching styles, then partition the entire database into smaller manageable subsets. These subsets, which yield substantially faster classification times, deliver recognition accuracies that are on par with those achievable through the use of the entire database. We demonstrate the utility of our approach with empirical results obtained from databases of various sizes and characteristics.

For enrolling a new user into the system, we propose a model that identifies sketching style of the new user by taking only a few sample adaptation data. After recognizing the style, instead of using a bulky model trained by all data to recognize new sketches of that user, we use recognized style model which yields considerably small classification time with respect to former model.

## 1.4 Thesis Roadmap

In the rest of this paper, we first describe our resource-aware training method based on style adaptation and then sketching style identification method for new users. Our main contribution is a practical, simple and easy to implement method, as evident from the brevity of the section describing our method. The bulk of the thesis is devoted to experiments illustrating the utility of our method. We evaluate our approach in three different scenarios by measuring performance with respect to recognition accuracy, training and classification time. After that, we analyze how style-based groupings works as expected and then we conclude evaluation part with a section that reports success rate of the style identification method. At the end, we talk about related work and a summary of future research directions.

Chapter 2

# STYLE-BASED GROUPING OF TRAINING DATA

## 2.1 Introduction

Our approach for achieving shorter training times is based on the key observation that forming groups of users with similar sketching styles, and training a separate model for each group significantly reduces training times without sacrificing accuracy. Furthermore, classification time performance of these obtained style models notably better compared to the bulky model trained by using all training data. Hence, grouping users into a hierarchy based on the similarity of their sketching styles is a key component of our overall strategy.

All experiments are conducted by using Support Vector Machines (SVMs). We use grid search to find optimum parameters (-c and -g) of the SVMs. For feature representation, Image Deformation Model (IDM) features are used. All results are generated for two different sketch domain databases which will be discussed in more details in the database section. Throughout the experiments, all training and testing data parts are separate. For example, in the next section, each time we extract a user's data from the whole dataset to allocate it for testing data and the remaining for training data.

## 2.2 Generating Similarity Matrix

The term *sketching style* lacks precise definition and is subject to interpretation. Rather than attempting to define it formally, we adopt an operational definition. In particular, we say that two people have similar styles to the degree that a classifier trained with one user's data can accurately classify the other user's sketches. As we show later, this simple operational definition not only appeals to intuition, but also matches the colloquial sense of the term as demonstrated with real examples.

Let $S_{ij}$ be the accuracy of a classifier trained with data from user $i$ on the data of user $j$, and $S_{ii} = 0$. We set $S_{ii}$ to the zero because it has no meaning to talk about similarity of somebody with himself. Also, agglomerative hierarchical clustering method which takes similarity matrix as input, expects zero-diagonal of the similarity matrix. In effect, if there are $M$ users, $S$ represents an $M \times M$ similarity matrix as seen in Fig.2.1 where higher values in the matrix correspond to user pairs with similar drawing styles.

Fig.2.2 shows the multi-dimensional scaling (MDS) demonstration of similarity matrices for both databases. MDS is a method of visualizing the level of similarity of individual cases of a dataset. Each point represents an individual in a dataset and different colours indicates distinct style clusters that will be discussed in more details in the next section.

## 2.3 *Agglomerative Hierarchical Clustering*

After the generation of similarity matrix, the style-based grouping of the data is obtained by applying a standard linkage algorithm (e.g., the `linkage` function in Matlab) to build the corresponding hierarchical cluster tree from the similarity matrix $S$.

Since output of the linkage function is a hierarchical cluster tree, we should apply one more operation to construct clusters. Again, we use standard cluster function (e.g., the `cluster` function in Matlab) which cuts the hierarchical cluster tree by using any user-defined threshold or user can specify the exact number of clusters to the function beforehand.

The goal of constructing clusters is to partition all-data into smaller subsets. Due to these subsets, we can be able to generate simpler models and it helps to reduce training and classification times. However, as mentioned before, one of our main objective is not to significantly sacrifice with recognition accuracy rate. If we divide all-data overmuch, then we can end up with too much smaller recognition accuracy rates because of inadequacy of data even if we gain in terms of training and classification times. Therefore, while we determine clusters from a hierarchical tree, we always take the balance between accuracy and the time into consideration. Although, in literature, there are some proposed methods to find optimal number of clusters in hierarchical clustering [Jung et al., 2003], we can say that none of these approaches are definitive since the interpretation of the resulting hierarchical structure is context dependent. Therefore, we define the clustering stopping criteria particular for our objectives. During cluster formation, if one of the clusters suffers from inadequancy of data which means that consisted of very few users's data, then we stop dividing the data into subsets anymore.

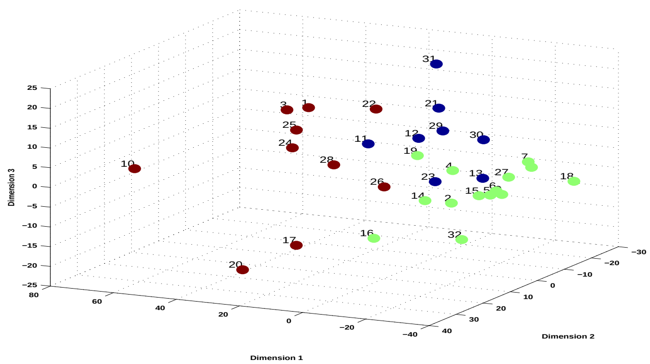
Fig. 2.3 shows results of the hierarchical clustering performed by linkage analysis. As the dendrogram structure of the hierarchy illustrates, it is possible to partition both databases into progressively smaller clusters until we reach individual users. Obviously, if the number of clusters well exceeds the number of user styles genuinely present in the database, the clusters cease being useful.

As mentioned, the simplicity of our method for style-based grouping of training data is deliberate. It can easily be implemented by a graduate or an undergraduate student with minimal programming and machine learning background.

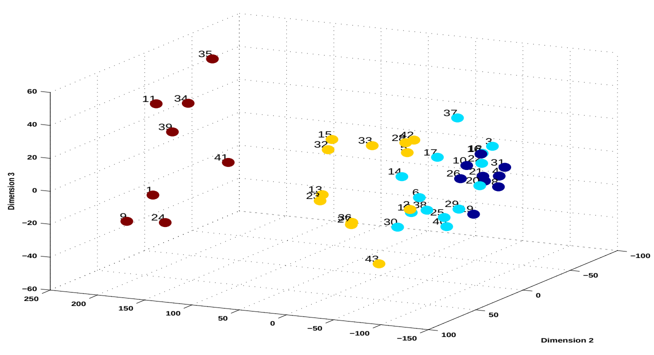
At the end of this step, we have multiple style based sub-groups by dividing the all available training data. Owing to these smaller data groups, we are able to reduce total spent training time and instead of having only one bulky model, generate multiple nimble system models whose classification time performances are also higher.

| Test with / Train with | User 1 | User 2 | User3 | ... | User M |
|---|---|---|---|---|---|
| User 1 | 0 | $S_{12}$ | $S_{13}$ | ... | $S_{1M}$ |
| User 2 | $S_{21}$ | 0 | $S_{23}$ | ... | $S_{2M}$ |
| User 3 | $S_{31}$ | $S_{32}$ | 0 | ... | $S_{3M}$ |
| ... | ... | ... | ... | 0 | ... |
| User M | $S_{M1}$ | $S_{M2}$ | $S_{M3}$ | ... | 0 |

Figure 2.1: Structure of a Similarity Matrix. $S_{ij}$ be the accuracy of a classifier trained with data from user $i$ on the data of user $j$, and $S_{ii} = 0$.

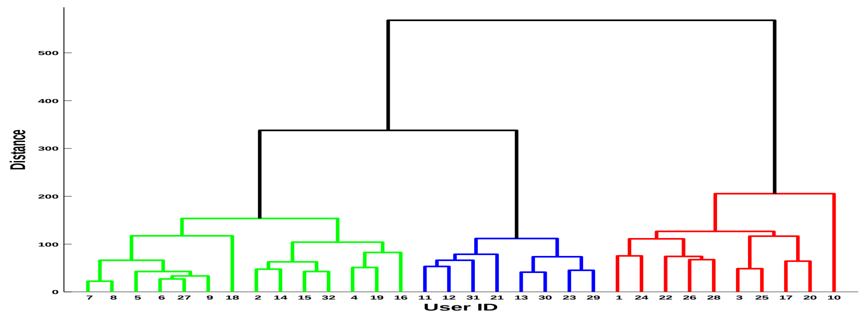



(a) NicIcon Database




(b) Traffic Database

Figure 2.2: Visualization of similarity of individuals by using matlab function `mds`.




(a) NicIcon Database


(b) Traffic Database

Figure 2.3: Results of the hierarchical clustering performed by linkage analysis. As the dendrogram structure of the hierarchy illustrates, it is possible to partition both databases into progressively smaller clusters until we reach individual users.

Chapter 3

# SKETCHING STYLE IDENTIFICATION OF NEW COMING USER

## *3.1 Introduction*

In order to integrate a new coming user into the clusters system explained in *Style-Based Grouping of Training Data* section, first we have to decide that which style cluster the new user belongs to. To be able to accomplish this, we collect some adaptation data from these users. Since long adaptation processes can be frustrating experience for users, at this point our main objective is to shorten the adaptation step as much as possible. Therefore, our style identification system needs to take just one sample sketch from each object class to determine style cluster of a new coming user. For example, since there are 8 different object classes in IUI Traffic Database, taking 8 sample sketches in total was enough to assign a new user to his style cluster.

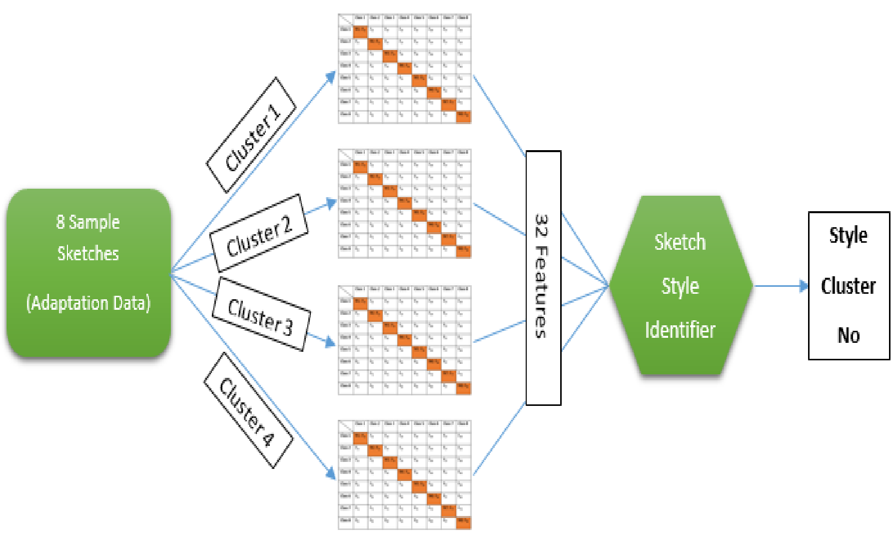
## *3.2 Methodology*

One of the main features of LIBSVM is to provide probability estimates for object classes. When we configure required parameters (e.g., the `-b parameter` in LIBSVM), LIBSVM gives probability values as to how much a given sample resembles each and every one of object classes. Intuitively, we propose that user's own style cluster tends to give large probability estimates for objects's correct classes compared to other clusters and these probability values show a learnable pattern. With the help of LIBSVM, we trained a style predictive model by using probability estimates as training data.

Fig. 3.1 shows feature extraction and style identification procedure for a new coming user to the IUI Traffic database. Since there are 8 different object classes in this database, we take 8 sketch samples from the user as adaptation data. (Note that one sample for each different object class) After that, we give the adaptation data to a style cluster model as input and get 8x8 probability matrix as seen in Fig. 3.1a. Let $P_{ij}$ be the probability of assigning a given sample sketch from object class i to object class j. Instead of using all probability values, we just extracted the diagonal ($P_{ii}$'s) of this matrix. Fig. 3.1b shows the overview of our style identification system. The adaptation data is given to each cluster model as input and probability matrices are generated independently each time. At the end of this step, we collect diagonal probability values of each matrix and establish the feature set for the user. As we stated before, since we have 4 different style clusters for IUI Traffic database, we end up with 32 features eventually. This feature set is given to the system which is previously trained with labelled data and get the cluster number for the new coming user as final output.

| | Class 1 | Class 2 | Class 3 | Class 4 | Class 5 | Class 6 | Class 7 | Class 8 |
|---|---|---|---|---|---|---|---|---|
| Class 1 | f#1: $P_{11}$ | $P_{12}$ | $P_{13}$ | $P_{14}$ | $P_{15}$ | $P_{16}$ | $P_{17}$ | $P_{18}$ |
| Class 2 | $P_{21}$ | f#2: $P_{22}$ | $P_{23}$ | $P_{24}$ | $P_{25}$ | $P_{26}$ | $P_{27}$ | $P_{28}$ |
| Class 3 | $P_{31}$ | $P_{32}$ | f#3: $P_{33}$ | $P_{34}$ | $P_{35}$ | $P_{36}$ | $P_{37}$ | $P_{38}$ |
| Class 4 | $P_{41}$ | $P_{42}$ | $P_{43}$ | f#4: $P_{44}$ | $P_{45}$ | $P_{46}$ | $P_{47}$ | $P_{48}$ |
| Class 5 | $P_{51}$ | $P_{52}$ | $P_{53}$ | $P_{54}$ | f#5: $P_{55}$ | $P_{56}$ | $P_{57}$ | $P_{58}$ |
| Class 6 | $P_{61}$ | $P_{62}$ | $P_{63}$ | $P_{64}$ | $P_{65}$ | f#6: $P_{66}$ | $P_{67}$ | $P_{68}$ |
| Class 7 | $P_{71}$ | $P_{72}$ | $P_{73}$ | $P_{74}$ | $P_{75}$ | $P_{76}$ | f#7: $P_{77}$ | $P_{78}$ |
| Class 8 | $P_{81}$ | $P_{82}$ | $P_{83}$ | $P_{84}$ | $P_{85}$ | $P_{86}$ | $P_{87}$ | f#8: $P_{88}$ |

(a) Probability matrix generated from each style cluster.




(b) Overview of Style Identification System

Figure 3.1: Style Identification System Overview for New Coming User

Chapter 4

# EVALUATION AND DISCUSSION

## 4.1 Introduction

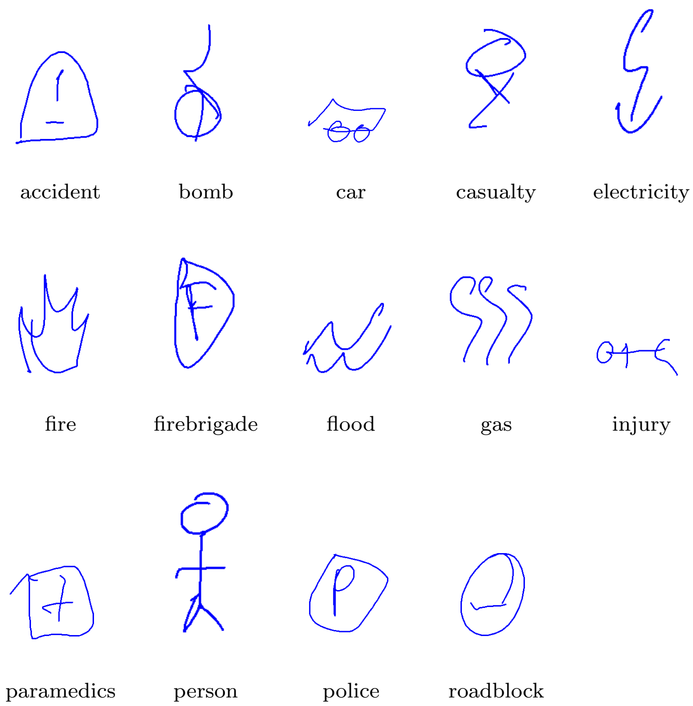
Our method is simple. Yet it can effectively divide training data into user groups with similar style, which can then be used to build highly accurate but simpler classifiers within less time. Hence, our evaluation focuses on these three main issues: accuracy, training time and classification time of the generated style models. In the following sections, we introduce the dataset used in our experiments, define a baseline, and then report results from experiments. Other than the results, we also have a section to evaluate the ability to perform style-based grouping. All statistical results that we report adopt a p-value of 0.05 unless otherwise noted.

## 4.2 Databases

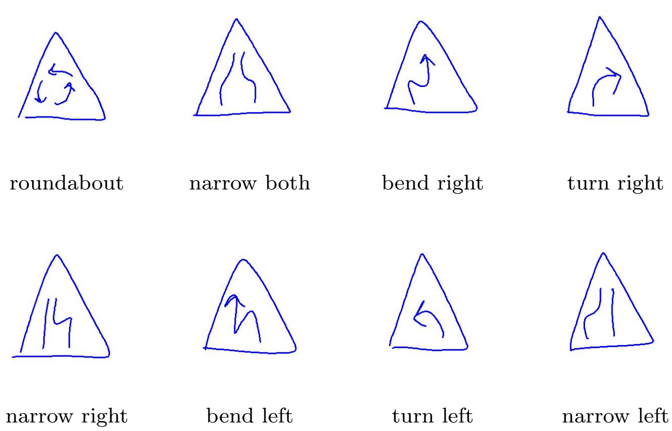
All our experiments were performed with two hand-drawn datasets, the NicIcon Database [Niels et al., 2008], and the IUI Lab Traffic Sign Database [Ozen and Sezgin, 2013]. The NicIcon database contains a total of 24,441 symbols from the crisis management domain (14 classes, 32 users, 55 sketches per class per user). The IUI Lab Traffic Sign Database contains a total of 10320 traffic sign sketches (8 classes, 43 users, 30 sketches per class per user). Representative examples from these databases can be found in Fig. 4.1.

## 4.3 Experiments

In order to evaluate the performance, we compared the accuracy of our system to the randomly selected method, baseline and the upper limit accuracies obtained for




**(a) NicIcon Database**




**(b) Traffic Signs Database**

Figure 4.1: Representative examples from the NicIcon and Traffic Signs databases.

the NicIcon and the Traffic Sign databases. All recognition results reported in this section were obtained by Support Vector Machines with RBF kernels using Image Deformation Model features [Ouyang and Davis, ]. Our choice of SVMs is motivated by their established success in sketch recognition as well as the excellent balance they strike between resource usage and accuracy [Ulaş et al., 2012].

In our experiments, training sessions were carried out using 5-fold cross validation with separate training and test datasets. Each training session included a grid-search to estimate hyperparameters of the SVM.

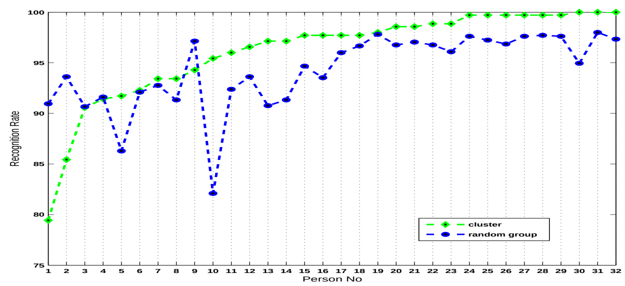
The main issue that we are addressing is the overly long training periods and poor classification times performance. Hence, we have also analyzed the time requirements for our method, as well as the randomly selected subsets, the N-nearest users and the all-data strategies.

### *4.3.1 Style Clusters vs Randomly Selected Groups*

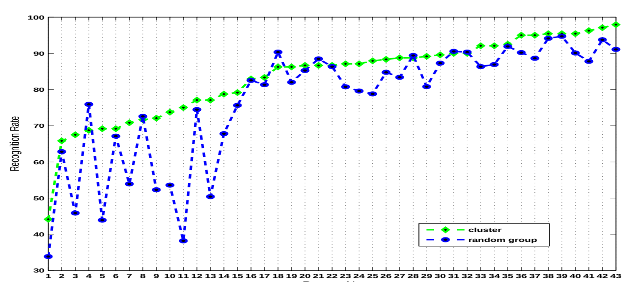
The crucial point of our approach is to decrease training data size by taking advantages of style groupings. In order to verify that our style-based system performs better compared to randomly selected training data groups, for the each member of clusters, we form random training data subsets which has approximately same size with their style clusters. It is a well-known fact that training time and complexity of any SVM system model increase with the enlargement of training data size. Since training data size of the two groups are approximately same, at this point it is unnecessary to make a comparison between time performances of these groups. However, in terms of accuracy, superiority of our proposed method over randomly selected groups can be seen in Fig. 4.2.

### *4.3.2 Style Clusters vs Baseline Method*

We implemented an alternative algorithm to serve as a baseline. The baseline method attempts to reduce training and classification time of a model by learning individual recognizers for each user with data from the N-nearest users. This method starts




(a) NicIcon Database




(b) Traffic Database

Figure 4.2: Comparison of recognition accuracies between clusters and randomly selected groups.

by building the similarity matrix $S$ as our method does, and for each user, trains a user-specific model. This method is as easy to implement as our method, hence it serves as an appropriate baseline.

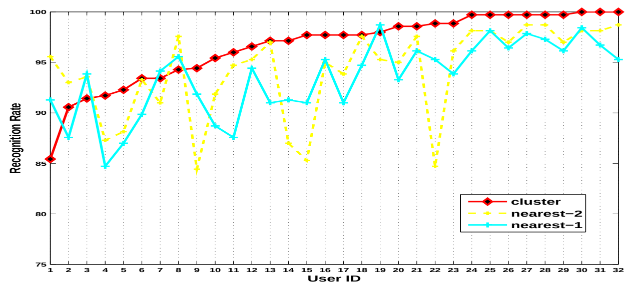
*Accuracy Comparison*

Fig. 4.3 shows the cluster accuracies, and the baseline accuracies obtained by models trained on data from the N-nearest users for $N = 1, 2, 3$. We have not computed accuracies for larger values of $N > 2$, and $N > 3$ for the NicIcon and the Traffic Sign databases respectively because the baseline requires more time for training compared to our method (see Fig. 4.4). As seen in these figures, the baseline accuracies are substantially worse than the cluster accuracies. The differences are statistically significant for both databases ($p = 0.0043$ for NicIcon for $N = 2$ and $p = 0.043$ for the Traffic Sign Database for $N = 3$. In all cases the p-values are smaller for $N < 2$ and $N < 3$ for the NicIcon and the Traffic Sign Database respectively).

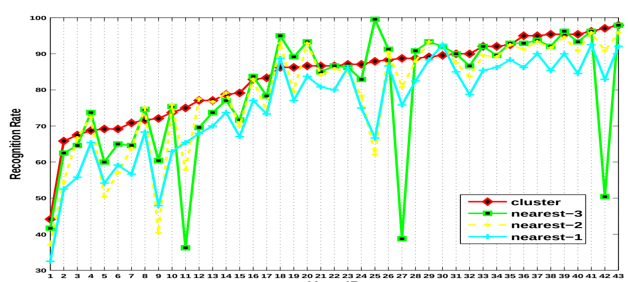
*Time savings analysis*

As we stated before, classification time of a system model depends only the size of training data being used in learning period of this model. For any user, therefore, if we select nearest-N users where size of N is less than the user's cluster size, classification time performance of baseline method will be higher compared to our style-based groupings approach. However, the main issue that we are addressing is while decreasing training and classification times, we also still want to remain in a reasonable recognition accuracy range. In preceding section, comparison with cluster accuracy rates shows how nearest-N users method failed against our cluster approach. Besides accuracy, experiments showed that our method also performs well in terms of training time compared to baseline. Hence, we have also analyzed the training time requirements for our method as well as N-nearest users strategies.

Fig. 4.4 shows the time requirements for both two methods. As seen in this figure, the baseline method actually has worse time requirements for $N > 2$, and

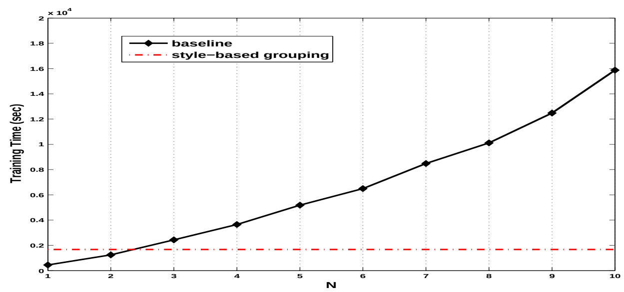



(a) NicIcon Database

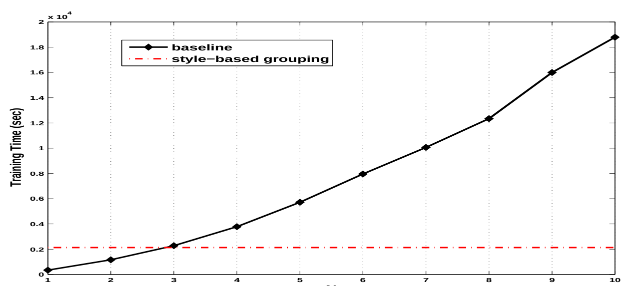



(b) Traffic Database

Figure 4.3: Comparison of the style-based groupings and baseline accuracies for the NicIcon, and the IUI Lab Traffic Sign databases. Users are sorted by their ascending style-based groupings score. For all values of $N = 1, 2, 3$, the baseline performs worse than the our style cluster approach, and the difference is statistically significant.




(a) NicIcon Database




(b) Traffic Database

Figure 4.4: Training time comparison of style-based groupings with nearest-N method.

$N > 3$ for the NicIcon and the Traffic Sign databases respectively. For values of $N$ where the time performance is good, the accuracies suffer significantly as summarized in the previous section, and in Fig. 4.3. So, the baseline fails to strike a balance between saving training/classification time and preserving accuracies. Our style-based grouping method, however, enables reasonable classification times and faster training process without sacrificing recognition accuracies.

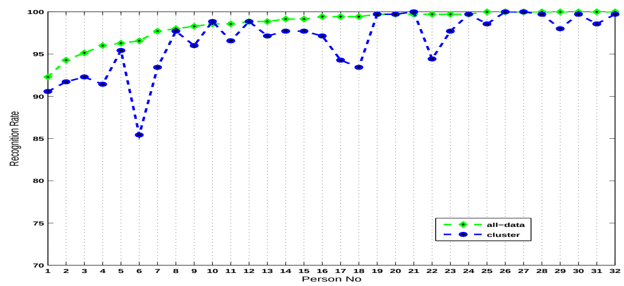
### *4.3.3 Style Clusters vs Upper Limit*

In order to evaluate effectiveness of building classifiers with style-specific subsets of the training data, we ran a series of experiments where we compared the recognition accuracies of models trained by style-specific subsets to those obtained by classifiers trained on all data. Because we would expect to get the best accuracies with classifiers trained on all available data, the all-data accuracies serve as upper limits on recognition accuracies. Our goal is not to surpass these values, but to come sufficiently close.

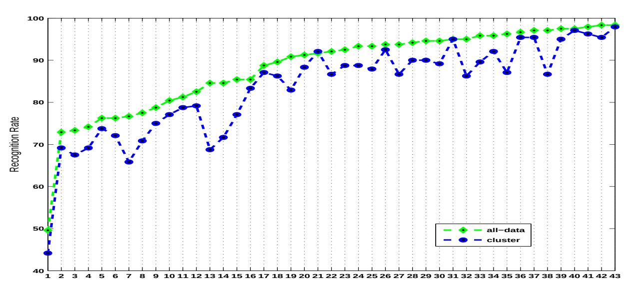
#### *Accuracy Comparison*

Fig. 4.5 shows the accuracies obtained by our style-based grouping method, and the all-data accuracies obtained by recognizers trained with all available data in the NicIcon and IUI Lab Traffic Sign databases. NicIcon is divided into three and Traffic Sign Database is divided into four groups based on style. The y-axes in the figures show accuracies. Users are sorted along the x-axis by their all-data accuracies in an ascending order (i.e., the x-axis value denotes the user with the $i^{th}$ lowest all-data accuracy).

The all-data accuracies serve as upper limits, which we do not expect to exceed. As seen in these graphs, our method compares favorably against the upper limit figures. Results of a t-test on the observed values did not find the differences to be significant.




(a) NicIcon Database

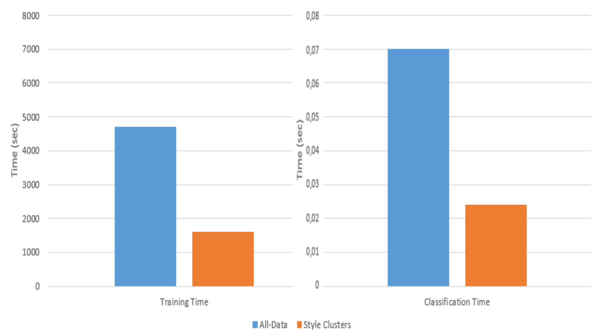



(b) Traffic Database

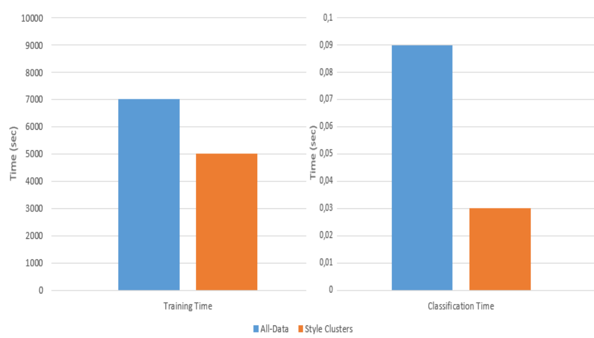
Figure 4.5: All-data accuracies, and accuracies obtained by our style-based grouping method. The all-data accuracies serve as upper limits. The difference was not found to be statistically significant. Users are sorted by their ascending all-data score.

*Time savings analysis*

At the beginning of the project, instead of beating accuracy rates of all-data model, we defined our ultimate goal as making significant improvements on training and classification times. In fact, as seen in Fig.4.6, we reached our goal by decreasing training and classification times one third of all-data model's time values.




(a) NicIcon Database




(b) Traffic Database

Figure 4.6: Training and Classification Time Comparison of style-based groupings and all-data methods.

## 4.4 Analysis of Groupings

Our resource-aware approach to training sketch recognizers via style adaptation relies on the assumption that grouping data based on user styles would result in accuracies comparable to the all-data values with less time requirements. Results in section 4.3

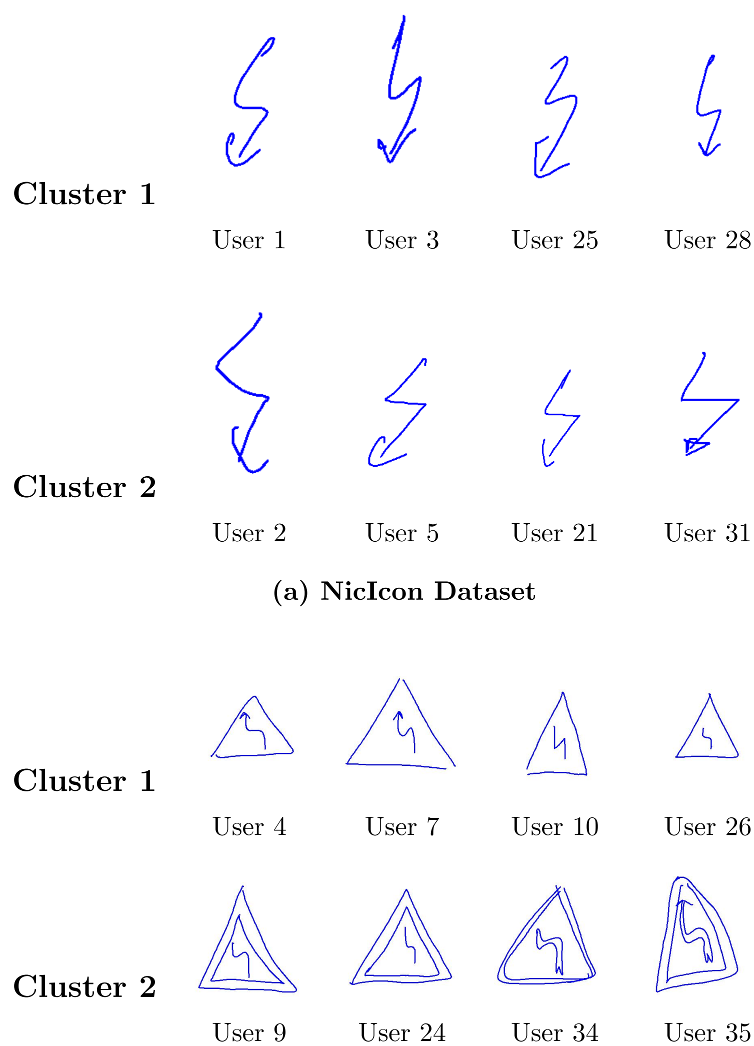
show that we can indeed match all-data accuracy performance with less training time and by means of generating simpler models with better classification performance. In order to see how well the groups found by our algorithm reflect distinctive sketching styles, we inspected the nature of the data in the clusters in close detail.

*Real Examples of Style Clusters*

Our analysis of the groupings found by our method shows that clusters indeed correspond to distinct sketching styles. Fig.4.7 shows examples of drawings of users from clusters found by our algorithm. The intra-cluster similarity and inter-cluster differences in the drawing styles are clearly visible. For example, a group of users in the first cluster of the NicIcon database draw parts of the “electricity” symbol with rather round corners, while users in the other cluster prefer sharp corners. Similarly, for the Traffic Sign database, users in the fist cluster prefer single triangular boundaries, whereas users in the second cluster adopt double lines to emphasize borders.

In fact, the similarity of sketching styles persists further down in the clustering hierarchy, much below the top level. For example, users 10 and 26 in the first cluster of the Traffic Sign database omit arrow caps on the “bend left” symbol, while users 4 and 7 strictly include arrow caps (see Fig. 4.7b, cluster 1). These users share the same clusters as deep as the $4^{th}$ level of the hierarchy from the top (see Fig.2.3b). Similarly, users 9 and 24 draw the “bend left” curve with a single curve, while users 34 and 35 use double lines to indicate the ’S’ shaped curve. These users are also placed close together in the hierarchy (see Fig. 2.3b).

These findings not only show the existence of distinct sketching styles, but also demonstrate our method’s ability to exploit these styles. Our method is based on a simple operational definition of sketching style, and is capable of finding groupings that agree with the intuitive and colloquial sense of the term.




**(a) NicIcon Dataset**

**(b) Traffic Sign Database**

Figure 4.7: Examples of drawings of users from clusters found by our algorithm

## 4.5 Equal Number of Support Vectors

Complexity of a sketch recognition model trained with an SVM corresponds closely to the number of selected support vectors. In a general sense, we can assert that increase in training data size is directly proportinal to number of selected support vectors for a model. The more support vectors exist in an SVM model, the higher it achieves recognition accuracy. Therefore, behind the reason of obtaining higher accuracy rates by using all-data compared to our system is because of larger training data size.

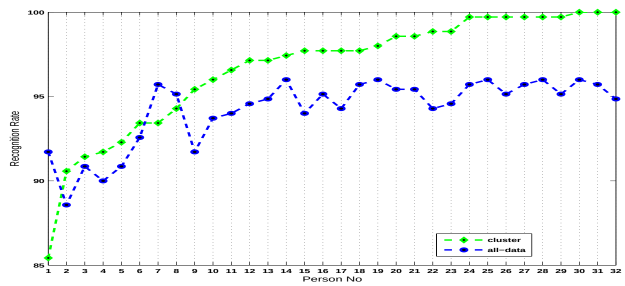
In order to qualify success of style-based grouping, we set up an experiment in a manner of equalizing number of support vectors between all-data model and style cluster models. Thus, we have a chance to compare accuracies of these models on an equal footing. For this purpose, we adjust the *-nu* parameter of LIBSVM which enables us to regulate the number of support vectors in an SVM model.

Fig. 4.8 shows the accuracies obtained by our style cluster models and the all-data accuracies taken from the recognizer trained with all available data. Under the equal support vectors condition, as seen in the figure, recognition accuracies are significantly better acquired by style-based groupings compared to all-data model.

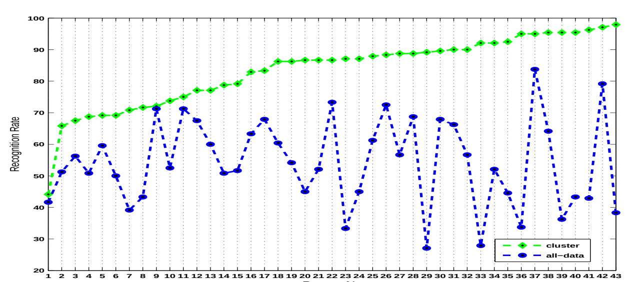
## 4.6 Analysis of Style Identification of New Coming User

In order to analyze style identification of new coming user, we try to explain all steps over one of the databases which is Traffic Sign Database. As we stated before in databases section, in IUI Traffic Database, there are 8 different object classes and 30 samples collected from one user for each object class. Since we need to take one sample for each class to assign the new user into his style cluster, we divided each existing user's data into 30 different test packages each of which is composed of 8 different samples.

For each one of the users, we extract all data of that particular user from the dataset as if he is an incoming person and has not provided any data yet. After that, each of his 30 packages is given independently as test data into the style identification




(a) NicIcon Database

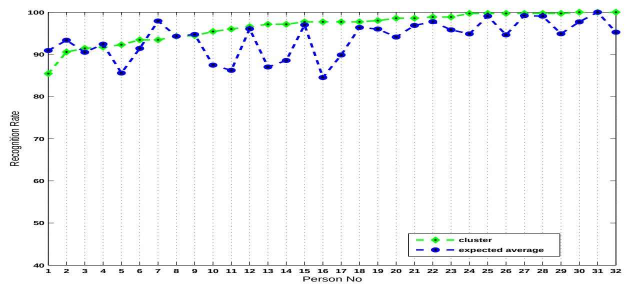



(b) Traffic Database

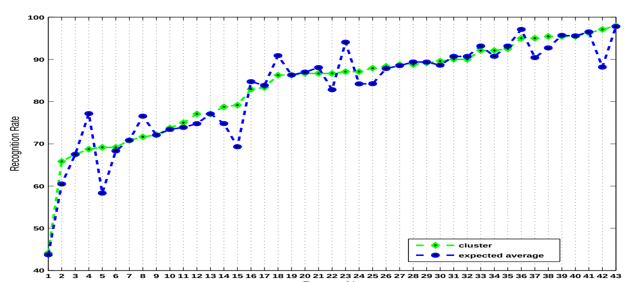
Figure 4.8: Accuracy comparison between style-based groupings and all-data method under the same number of support vectors case.

model which is previously trained with all others' data except the particular user. As a result of this, the model outputs 30 different cluster numbers for all packages. Therefore, we come up with a term *expected accuracy* that represents average of the accuracy rates that models of the 30 predicted clusters give to the particular user's data. In Fig. 4.9, we compare expected accuracy of each user with the accuracy of their real clusters which is assigned when all-data of a user is available in the dataset. As seen in this figure, expected accuracies are close enough to the real clusters's accuracies which shows the remarkable attainment of the our style identification system.

In previous system models, as we stated before, we use IDM as feature representation method. IDM extracts total of 720 features from a single sketch sample. When the number of features are high, SVM requires more time to train a system model because of complexity. However, as shown in chapter 3, for the style identification system model, we use only diagonal probability values of the matrices as our features so that it results in much less number of features compared to IDM case. Therefore, the time for training style identification system model is comparably smaller than the any aforementioned system models.




(a) NicIcon Database




(b) Traffic Database

Figure 4.9: Accuracy comparison of style-based groupings with expected accuracies obtained from style identification system.

Chapter 5

# RELATED WORK

Our work contributes to the list of sketch recognition related algorithms that can be implemented by people with minimal knowledge of machine learning theory. In that respect, it has the same motivation as the $1 and $N Recognizers [Wobbrock et al., 2007] [Anthony and Wobbrock, 2012].

We are not the first ones to recognize resource limitations as a practical issue in building sketch recognizers. Tirkaz et al. recognize excessive memory requirements of training sketch recognizers with large datasets, and describe a memory conscious recognition architecture that preprocesses the training data to retain only a few representative templates per class [Tirkaz et al., 2012]. They also employ a carefully constructed recognition architecture consisting of cascades of classifiers to improve on recognition speed. Unfortunately, their overall approach to recognition is rather complex, as it consists of 7 steps including preprocessing, feature extraction, prototype selection, template selection, building the so-called first and second classifiers, and classifier combination. In contrast, our style-based grouping is simple enough to be implemented by a graduate or an undergraduate student with a minimal background in programming and machine learning.

Another piece of work that recognizes resource limitations during the training phase advocates an adaptive strategy for incremental learning of sketch recognizers [Zhengxing et al., 2004]. In this work, Sun et al. adopt an SVM-based approach for classification, and they save training time and storage space with very small loss of accuracy. They advocate the view that, rather than using all the available data to train a classifier at once, one can start with an initial classifier trained with a small subset of the data, and then progressively build more accurate classifiers. During this process, they retain only the support vectors of the existing classifier and add in new samples to build a new model. This work falls under the incremental learning category, and again requires substantially more effort to understand and implement compared to the simple approach that we present.

The strategy of detecting, and adapting to user styles has been explored in other areas such as speech recognition and handwriting recognition [Yamagishi et al., 2009], [Torbati et al., 2012], [Chellapilla et al., 2006]. Albeit, these have mainly focused on tuning classifiers with limited self-examples from users to achieve good accuracies. Of these, work by Chellapilla et al. on allograph based writer adaptation for handwritten character recognition is the most relevant piece of work [Chellapilla et al., 2006]. In this work, the authors throw away writer identity information, and perform a large scale clustering of the entire training data into groups of allographs. They later train classifiers to predict the allograph classes, and use a personalized SVM model to combine the allograph classifier with the limited examples of a new user to obtain a personalized classifier. In contrast, we explicitly use the identity of users during the construction of our similarity matrix, and perform a linkage analysis over the pairwise cost matrix. Furthermore, although the work of Chellapilla et al. targets writer adaptation, their approach works with the entire data, and does not address the overly long training time requirements issue.

Although our style identification system is the first work for sketch recognition area in the literature, some researches are conducted on hand-writing style differentiation and writer identification [Utkarsh et al., 2012]. These works captures the writer style by using writer-specific properties such as slant and loops. They generate specific feature set for the purpose and have to train a new model with labelled data with the aid of these features. In our case, we don't have a feature set to differentiate sketching styles and therefore, there is no need to train a new system model which costs us extra time and memory. Instead, we make use of some LibSVM properties and approach the problem in a probabilistic manner by just using existing style cluster models.

Chapter 6

# FUTURE WORK

Our approach for style-based grouping consists of two fundamental stages. First, an inter-user similarity matrix is constructed, and then a standard linkage algorithm constructs a hierarchy of style groups. In both of these steps, we have explicitly avoided sophisticated procedures in order to preserve ease of implementation. However, it is possible to employ more sophisticated methods in either of these steps. For example, rather than computing a pairwise similarity matrix, one could measure similarity of pairs, or tuples of users through pairwise classifiers (i.e., train a model with a pair of users, test with another pair). This modified procedure could potentially lead to more accurate style-based partitioning of the data.

Another improvement could be done on the clustering step. For example, it might be possible to use prior knowledge of user styles, and perform constrained [Tung et al., 2001], or semi-supervised clustering [Sugato et al., 2004], [Basu et al., 2002] in order to obtain more representative user-style clusters.

In addition to these extensions, which stick to the original two step approach, one can adopt more sophisticated iterative approaches for building clusters. For example, cluster formations can be interleaved with intermediate steps that measure pairwise similarity and/or distinctness of user styles. However, these more sophisticated methods should be carefully designed to keep computational costs down, as each iteration of the similarity computation will require more processing. Monte Carlo methods could be used to achieve efficiency, however incorporating such sophisticated methods defeats the main feature of our approach, which is ease of implementation.

For style identification system, an improvement could be done on decreasing size of the adaptation data. In our current framework, new user should provide one sample for each object class in a sketch database. However, we can conduct more detailed analysis about which sketch samples better represent the style of a user. In this manner, system may need to take just these samples as adaptation data instead of getting one sample per object class. Further, if we prove the existence of style representative samples, new specific feature set can be generated to yield sketching style of any user after examine the properties of these sketches.

Chapter 7

# CONCLUSION

At the very beginning of the project, we stated our problem statement clearly and it provided us with a fair road-map. Although sketch recognition studies have been developing rapidly in recent years, this brings new challenges to the field. One of the major challenges is the recent increase in number and size of the sketch databases. In fact, having huge training data is a desired situation for any intelligent recognition system that uses machine learning technologies in their training sessions. The more training data they have, the more accuracy rates they can get from their systems. However, there also exists a balance between accuracy rates and time. We primarily focused on this issue in a manner that is fair to both accuracy rates and time.

## *7.1 Contributions*

We have described a simple and easy to implement method for partitioning a large corpus of training data into disjoint partitions based on drawing style. Our method uses a standard hierarchical clustering method, hence it can be implemented with minimal knowledge of machine learning. Furthermore, the separation between the similarity matrix generation and the clustering steps allows practitioners to substitute in more sophisticated clustering algorithms for the well known linkage algorithm. The style-based clusters that we generate make shorter training times feasible without a significant loss of accuracy. Besides decline in training time, since the complexity of resulting style models is simpler compared to the one that is trained with using all data, great advances also made on classification time.

In addition to just dividing existing sketch database into several smaller subsets, we also wondered how a new coming user can be included into our system. At this stage of the project, our main objective was to take as few as possible adaptation data from a new user. At the end, we obtained a style identification system model that requires just one adaptation sample from each object class.

To sum up our contribution in this thesis, when a new user comes to the sketch recognition system, he should provide a few adaptation data to be assigned into any style clusters that are previously generated again due to our framework. After the style cluster of the user is identified via our style identification system, his new sketches are going to be recognized by using his style cluster model. In this way, we gain in terms of training and classification time without a significant loss of accuracy. We have demonstrated the effectiveness of our approach with empirical results obtained from two databases of hand drawn symbols. Finally, we showed that groups generated by our method do indeed capture sketching styles in the intuitive sense of the phrase.

## 7.2 Discussion

We try to keep our method as simple as possible. Being simple is the most distinctive feature of our system compared to all similar works. Although, there are some effective machine learning algorithms such as Support Vector Machines that we used in this project, the problem of these algorithms is to spend too much time while processing the data. Therefore, we determined our success criteria as how much we gain in terms of time while we are still having reasonable recognition accuracy rates. The results in chapter 4.3 demonstrates that due to the our method, we get simpler cluster models instead of a big complex one in less training time. These simpler models are also able to provide fast classification on new test samples. On the other hand, although we loss trainining data because of dividing whole data corpus into smaller subsets, recognition accuracy rates do not decrease significantly according to paired t-test results.

The section 4.6 shows that how successful our system identifies the style of a new user despite small adaptation data size. This system also represents the first work in the literature that deals with style identification in a sketch recognition framework.

# BIBLIOGRAPHY


[Anthony and Wobbrock, 2012] Anthony, L. and Wobbrock, J. O. (2012). $n-protractor: a fast and accurate multistroke recognizer. In *Proceedings of Graphics Interface 2012*, GI '12, pages 117–120, Toronto, Ont., Canada, Canada. Canadian Information Processing Society.

[Basu et al., 2002] Basu, S., Banerjee, A., and Mooney, R. J. (2002). Semi-supervised clustering by seeding. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pages 27–34, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Chellapilla et al., 2006] Chellapilla, K., Simard, P., and Abdulkader, A. (2006). Allograph Based Writer Adaptation for Handwritten Character Recognition. In Lorette, G., editor, *Tenth International Workshop on Frontiers in Handwriting Recognition*, La Baule (France). Université de Rennes 1, Suvisoft. http://www.suvisoft.com Université de Rennes 1.

[Jung et al., 2003] Jung, Y., Park, H., Du, D. Z., and Drake, B. L. (2003). A decision criterion for the optimal number of clusters in hierarchical clustering. *J. of Global Optimization*, 25(1):91–111.

[Niels et al., 2008] Niels, R., Willems, D., and Vuurpijl, L. (2008). The NicIcon database of handwritten icons. *11th International Conference on the Frontiers of Handwriting Recognition (ICFHR 2008)*. In press.

[Ouyang and Davis, ] Ouyang, T. Y. and Davis, A. A visual approach to sketched symbol recognition.

[Ozen and Sezgin, 2013] Ozen, B. and Sezgin, M. (2013). IUI Traffic Sign Database. `http://iui.ku.edu.tr/IUITrafficDB`.

[Settles, 2009] Settles, B. (2009). Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

[Sugato et al., 2004] Sugato, B., Banjeree, A., Mooney, E., Arindam, B., and Raymond, J. (2004). Active semi-supervision for pairwise constrained clustering. In *In Proceedings of the 2004 SIAM International Conference on Data Mining (SDM-04*, pages 333–344.

[Tirkaz et al., 2012] Tirkaz, C., Yanikoglu, B., and Sezgin, M. (2012). Memory conscious sketched symbol recognition. *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 314–317.

[Torbati et al., 2012] Torbati, H. N., A.H., Picone, J., and Sobel, M. (2012). Applications of dirichlet process mixtures to speaker adaptation. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4321–4324.

[Tung et al., 2001] Tung, Anthony, K. H., Ng, R. T., Lakshmanan, L. V. S., and Han, J. (2001). Constraint-based clustering in large databases. In *Proceedings of the 8th International Conference on Database Theory*, ICDT '01, pages 405–419, London, UK. Springer-Verlag.

[Ulaş et al., 2012] Ulaş, A., YiLdiZ, O. T., and Alpaydin, E. (2012). Cost-conscious comparison of supervised learning algorithms over multiple data sets. *Pattern Recogn.*, 45(4):1772–1781.

[Utkarsh et al., 2012] Utkarsh, P., Chetan, R., Arti, S., and G., V. (2012). Structural learning for writer identification in offline handwriting. In *ICFHR'12*, pages 417–422.

[Wobbrock et al., 2007] Wobbrock, J. O., Wilson, A. D., and Li, Y. (2007). Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST '07, pages 159–168, New York, NY, USA. ACM.

[Yamagishi et al., 2009] Yamagishi, J., Kobayashi, T., Nakano, Y., Ogata, K., and Isogai, J. (2009). Analysis of speaker adaptation algorithms for hmm-based speech synthesis and a constrained smaplr adaptation algorithm. *Audio, Speech, and Language Processing, IEEE Transactions on*, 17(1):66–83.

[Yann and Corinna, 2009] Yann, L. and Corinna, C. (2009). The MNIST DATABASE of handwritten digits. `http://yann.lecun.com/exdb/mnist/`.

[Zhengxing et al., 2004] Zhengxing, S., Shuang, L., and Lisha, Z. (2004). Sketchy shape recognition based on incremental learning. *Workshop on Intelligent Virtual Environment and Mixed Reality, 14th International Conference on Artificial Reality and Teleexistence of(ICAT 2004)*, pages 314–317.