Feature Extraction and Classifier Combination for Image-based Sketch Recognition

R. Sinan Tumen[1], M. Emre Acer[2] and T. Metin Sezgin[1]

[1]Koc University, Istanbul, Turkey
[2]Carnegie Mellon University, Pittsburgh, PA

Abstract
Image-based approaches to sketch recognition typically cast sketch recognition as a machine learning problem. In systems that adopt image-based recognition, the collected ink is generally fed through a standard three stage pipeline consisting of the feature extraction, learning and classification steps. Although these approaches make regular use of machine learning, existing work falls short of presenting a proper treatment of important issues such as feature extraction, feature selection, feature combination, and classifier fusion. In this paper, we show that all these issues are significant factors, which substantially affect the ultimate performance of a sketch recognition engine. We support our case by experimental results obtained from two databases using representative sets of feature extraction, feature selection, classification, and classifier combination methods. We present the pros and cons of various choices that can be made while building sketch recognizers and discuss their trade-offs.

Categories and Subject Descriptors (according to ACM CCS): Information Interfaces and Presentation [H.5.2]: User Interfaces—Evaluation/methodology, Interaction styles, Prototyping, User-centered design Computing Methodologies [I.5.4]: Pattern Recognition—Applications

1. Introduction

Image-based approaches to sketch recognition encompass a large portion of the existing sketch recognition algorithms in the literature. Unlike temporal and constraint-based approaches, which generally model measurements taken between stroke pairs/groups, image-based approaches treat sketches as raster images and compute fixed-length feature vectors for image patches. This property makes them well suited for the standard three-step machine learning pipeline, where first feature vectors are extracted from a set of example images, then classifiers are trained using these features, and then new sketches are classified using the trained models.

For image-based recognition methods, the issue of how sketches are converted to feature vectors is of critical importance, because the accuracy of the classification system ultimately depends on how well the features can capture important characteristics of the training data. In an effort to achieve better recognition accuracies, authors have attempted to adapt various feature extraction methods from the machine vision literature to the domain of sketch recog-

nition, and a number of these methods were shown to work well for hand-drawn data (e.g., Zernike Moments [KH90], IDM features [OD09], shape contexts [BMP02]). These features were then used within the three-step machine learning pipeline as explained above.

Although many feature extraction methods have been proposed in the literature, important issues regarding their practical use have not been discussed. In particular the following issues have not been addressed with sufficient detail:

1. To what extent do the recognition accuracies depend on parameter extraction settings?
2. What courses of action can be taken if an exhaustive search of feature extraction parameters is not feasible?
3. Do different feature extraction methods (e.g., shape contexts, Zernike moments) capture different and potentially complementary information?
4. If this is the case, can the complementary information be fused to improve recognition accuracy?

In this paper, we show that selecting good feature extraction parameters is important for achieving good sketch recognition accuracies. However, an exhaustive search of all

combinations of parameters is not practical in most cases. We show that, for cases where searching for optimal feature extraction parameters is not feasible, higher recognition accuracies can be obtained by training ensembles of classifiers on the same data. In particular, we illustrate that combining different classifiers yields higher recognition rates even for suboptimal feature sets.

We illustrate that when multiple feature extraction methods are available (e.g., shape contexts, Zernike moments etc.), recognition accuracies can be improved by performing feature-level fusion. Furthermore, using a feature-selection method (e.g., Maximum Relevance & Minimum Redundancy) yields further accuracy improvements, and also speeds up online recognition. These results also illustrate that not all features generated by feature extraction methods are useful, and filtering out irrelevant features improves recognition accuracy.

We also show that improvements due to feature selection come at the cost of substantial offline processing. We present quantitative evaluation results from two databases illustrating the trade-off between the offline processing required for feature selection, and the resulting gains in online speed and accuracy.

Over the course of evaluating feature extraction methods, we introduce the Extended Trace Transform (ETT), an extension of the Trace transform [KP01], which can deal with the free-hand nature of sketches. We show that ETT contributes many useful features toward the construction of optimal feature sets during feature selection. This demonstrates the utility of ETT as a compact and informative feature extraction method.

In the rest of this paper, we introduce components of the experimental framework that we use to explore how recognition accuracy changes with respect to factors including the choice of feature extraction parameters, feature selection, feature combination, and classifier fusion. Section 2 briefly introduces the feature extraction algorithms that we study including Zernike moments, IDM, shape contexts and ETT. Then we introduce two databases used in our experiments, and the list of classification methods used in our study. In Section 5, we report the results of a series of experiments that measure the effects of the aforementioned issues regarding feature extraction, feature selection, feature combination, and classifier fusion.

2. Feature Extraction Methods

A number of image-based feature extraction methods have been suggested in the literature. Here we focus on five methods. We use IDM and shape contexts, because they represent the state of art in terms of high recognition accuracy. We use two Zernike-moment-based features (Zernike moments and extended Zernike moments), because they are simple and easy to implement. We also introduce a new feature based

on an extension of the Trace transform [KP01]. Below we introduce each of these methods, and list their free parameters. Because trying out a large number of parameter combinations is impractical, for the purposes of our experiments, we limited the set of values to be tested to reasonable ranges, which when extended further on either direction cause substantial accuracy reductions.

2.1. Zernike moments

Zernike moments were demonstrated as simple and effective feature representations for sketch recognition [HN04]. The way we use Zernike moments is similar to the framework introduced by Hse et al., and we refer the reader to their paper for details [HN04].

We also use a variant of the Zernike moments, where we apply the feature extraction algorithm to the image data as opposed to the interpolated stroke data as described in [HN04]. We refer to these features as the *extended Zernike* features.

Both variants of Zernike moments have one free parameter, which is the order of the Zernike moment o. We used $o = \{3,4,5,...,21\}$, which results in a total of 19 different feature extraction parameters.

2.2. Image Deformation Model Features (IDM)

IDM features are adaptations of the annotated images, which were initially suggested by LeRec for handwriting recognition, to the domain of sketch recognition. This image-based feature representation was proposed by Ouyang et al. in [OD09].

IDM features have 720 entries, and the feature extraction mechanism has three free parameters, k (kernel size), σ (smoothing factor), and r (resampling parameter),. We set $k = 25$, and use the following candidate values for the other parameters: $\sigma = \{1,2,..,15\}$, $r = \{25,50,100,150\}$. This results in a total of 60 feature extraction parameters.

2.3. Shape Context

Shape context is a histogram based local descriptor which captures image intensity statistics in the neighborhood of a reference point. Oltmans et al. have adapted shape contexts to sketch recognition [Olt07].

Shape context has three free parameters, c (the number of concentric circles), s (the number of slices), r (radius of the shape context). We used the following candidate values for these parameters: $c = \{1,2,3,4\}$, $s = \{4,8,12,16,20,36\}$, $r = \{10,50,100,150,200\}$. These values result in a total of 120 different feature extraction parameters.

Oltmans et al. used a sophisticated matching method for using shape contexts to do image segmentation, however in
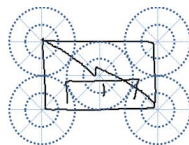
Figure 1: *Shape context placement.*

order to be able to fit them in our framework, we used a simpler approach and placed five shape context histograms on the corners and the center of the bounding box of each shape as illustrated in Fig. 1.

2.4. Extended Trace Transform (ETT)

ETT is our extension of the Trace transform, which itself is a generalization of the Radon transform [KP01]. ETT features for sketches are calculated from stroke data.

The feature extraction starts with a normalization step. In this step, first the symbol is scaled such that the standard deviations in horizontal and vertical directions are 1. Next, the scaled symbol is sampled into a 100x100 grid image. The points are translated so that the center of mass of the points is the center of the grid. Sampling of the symbol into the grid is performed such that the grid spans 2.5 standard deviations of the symbol in both directions. This normalization is similar to the process described in [OD09].

The resulting image $f(x,y)$ is the grid image of the symbol. This image is then convolved with a Gaussian filter $G(x,y)$ to obtain the smoothed image $g(x,y)$ according using Eq. 1 and Eq. 2.

$$G(x,y) = e^{-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2}\right)} \tag{1}$$

$$g(x,y) = f(x,y) * G(x,y), \tag{2}$$

The convolution is carried out to eliminate sampling errors of the stroke points and to smooth out the noise introduced during drawing. The convolved image g is rotated by discrete angles in the range [0-180]. For each angle θ, the columns of the image correspond to the image tracing lines $f(r,\theta,t)$ of the trace transform where r is distance of the tracing line to the origin and t is the parameter along the line. We apply a functional T on these tracing lines to obtain a single number for each image column. The result is a row vector for each angle θ. At the end of the rotations, the trace transform of the image is obtained using B number of bins and A number of angles. The result is an AxB image which is the extended trace transform of the sketch image using functional T.

We extract feature vectors from this transformed image

by applying the second functional, R, on the rows of the image. For each row, the functional R operates on the columns, producing a single value for each row. The final result is a feature vector of length B.

We have implemented and experimented with a total of seven functionals: *Radon*, *Harmonic*$_1$, *Harmonic*$_3$, FFT, *Integral*, *Fourier* and *Gradient.* The definitions of these functionals can be found in Table 1. A feature extraction operation consists of applying one functional (T) along the image tracing line and another functional (R) along image rows. Therefore, we had a total of 49 functional combination pairs to be evaluated for ETT. Of these 49 pairs, 4 corresponding to combinations of the first and the fourth functionals worked out best, and had comparable performance. Therefore we set T and R to the first functional in our experiments.

Setting the functionals leaves three free parameters to ETT: b (the number of bins), θ (the number of discrete angles), σ (the smoothing parameter). We used the following candidate values for these parameters: $b = \{20, 30, 40, 50\}$, $\theta = \{10, 20, 30, 40, 50\}$, $\sigma = \{1, 3, 7, 11, 15\}$. These values result in a total of 100 different feature extraction parameters.

3. Databases

We ran all our experiments using two databases. The first database includes symbols from our Course of Action Diagrams database, and the other database is the publicly available NicIcon database [Vuu].

3.1. Course of Action Diagrams Database (COAD)

The set of objects in this database come from the domain of military *Course of Action Diagrams* [Man97] shown in Fig. 2. A complete list of objects in this domain can be found in the US Army Field Manual 101-5-1. There are hundreds of symbols in this domain, and we focus on a subset of 20 for practical reasons.

Some symbols in this domain are quite distinct, while some others look similar. For example: the symbol for an *Enemy Artillery Observation Unit* (Fig. 2(d)) consists of the symbol for an *Enemy Observation Unit* (Fig. 2(g)) with a small circle in the middle.

Eight different users were shown symbols picked randomly and asked to sketch examples from each of the 20 symbol classes. In total 620 examples of different symbols were collected. The number of examples per symbol varied between 27 and 45, with a median of 30.

3.2. The NicIcon Database

The NicIcon database includes multi-stroke symbols that are used in the context of an emergency management application (e.g., symbols for fire brigade, gas, roadblock, injury

No	Functional	Definition	Details
1	$\int f(t)\,dt$	Radon transform	-
2	$\|\int f(t)e^{-jkw_0}\|$	Amplitude of the 1st harmonic	$k=1$
3	$\|\int f(t)e^{-jkw_0}\|$	Amplitude of the 3rd harmonic	$k=3$
4	$max(\|F\{f(t)\}\|)$	Fourier functional	F is the Discrete Fourier Transform
5	$[\int \|f(t)\|^p\,dt]^q$	Integral	$p=0.5, q=1/p$
6	$\int \|f(t)'\|\,dt$	Gradient functional	$f(t)'$ is the gradient of $f(t)$
7	$[\int \|F\{f(t)\}\|^p\,dt]^q$	Fourier	$p=4, q=1/p$

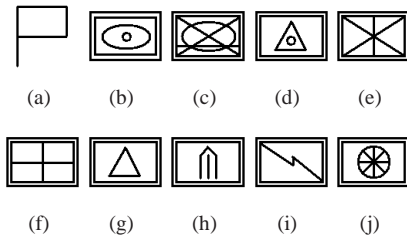**Table 1:** *Functionals for the Trace Transform.*

Figure 2: *A subset of symbols used in Course of Action Diagrams.*

Figure 3: *Examples of hand drawn Course of Action symbols used in our evaluation.*

[WNvGV09]). This database contains a total of 23641 symbols distributed into 14 classes.[†] The symbols in the database consist of an average of 5.2 strokes, and the average number of strokes for the individual categories ranges from 3.1 to 7.5.

[†] In order to keep our experiments tractable, we choose a smaller subset of 1400 symbols, 100 symbols from each class.

Figure 4: *Examples of hand drawn icons from the NicIcon database [WNvGV09]. From top-to-bottom and left-to-right, the symbols represent* fire brigade, gas, roadblock, injury, paramedics, police, accident, bomb, fire, car, person *and* flood.

4. Classifiers and Training Methodology

In our experiments we used ten different classifiers: 1. linear perceptron classifier, 2. k-nearest neighbor classifier, 3. Fisher's least square linear classifier, 4. Parzen classifier, 5. nearest mean classifier, 6. uncorrelated normal based quadratic Bayes classifier, 7. quadratic Bayes normal classifier, 8. linear classifier using PC expansion, 9. Parzen density based classifier, and 10. Linear Bayes Normal Classifier.

Unless otherwise mentioned, all our experiments were carried out using separate training, testing, validation sets, repeated 10 times on sets obtained through uniform sampling.

5. Experimental Setup and Results

In order to evaluate the effects of various design choices that one has to make during the design of a sketch recognizer, we set up a series of controlled experiments. Our first set of experiments measure the effects of feature extraction parameters on the recognition accuracy. Later experiments asses the suitability of classifier combination as a means of improving recognition accuracies even for suboptimal parameter settings. The last two sets of experiments are designed to assess the complementarity of various feature extraction methods, and the merits of combining the information they provide through feature fusion.

5.1. Feature Extraction

5.1.1. Effects of Optimal Parameter Selection for Feature Extraction

It is important to understand the sensitivity of recognizer accuracies to the values of various feature extraction parameters, because it serves as an indication of how well feature sets obtained through different parameter settings capture the important characteristics of sketches. These values also serve as an indication of the sensitivity of the results to the parameters of various feature extraction mechanisms.

As mentioned in our discussion of the feature extraction methods (Sec. 2), all feature extraction methods require a set of preset parameters. Using the candidate values mentioned in Sec. 2 gives us a large number possible feature extraction

Zernike	E.Zernike	Trace	IDM	Shape context
19	19	100	60	120

Table 2: *Number of feature extraction settings tested for each feature extraction method.*

settings to consider (reproduced in Table 2 for reference). For each of these settings, we measured the accuracy obtained using the set of 10 different classification algorithms for the Course of Action Diagrams (COAD) and NicIcon databases.

The solid blue lines in Fig. 5 show the variations in mean accuracies for each feature extraction method and database combination. As illustrated in these plots, there is substantial variation in accuracies across parameters.

It is important to note that searching for best set of parameters took time in the order of days, and may not be practical in general.

5.1.2. Classifier Combination with Suboptimal Parameters

Although we do not report the results in this paper due to space limitations, in the experiment above we also indirectly observed how well the 10 classifiers mentioned in Sec. 4 worked for all the different feature sets that were tested. Our tests have revealed that the performances of these classifiers were negatively correlated across feature sets. In other words, when some classifiers did poorly on a particular case, the others did relatively well, and vice versa.

Theoretical results from machine learning theory says that when this sort of behavior is observed, (i.e, when classifier accuracies are found to be negatively correlated), combining them yields higher accuracies [LY99]. Therefore a natural followup to the previous experiment is to test if it is possible to increase recognition accuracy with suboptimal feature extraction parameters if the resulting features are fed to ensembles of classifiers.

To test this, for each feature extraction method, each possible feature extraction setting, and each database, we measured the average accuracy of the 10 classification algorithms, and compared the average accuracies to those obtained by classifier combination. In our experiment, we used six different classifier combination schemes (product rule, mean rule, median rule, max rule, min rule and voting [KHDM98]).

The dashed red and solid blue lines in Fig.5 respectively show the mean accuracies for classifier combination methods and individual classifiers. As seen here, classifier combinations results in higher accuracies in all cases except the case of ETT on the NicIcon database. Wilcoxon's paired rank test for accuracies from all database & feature set com-
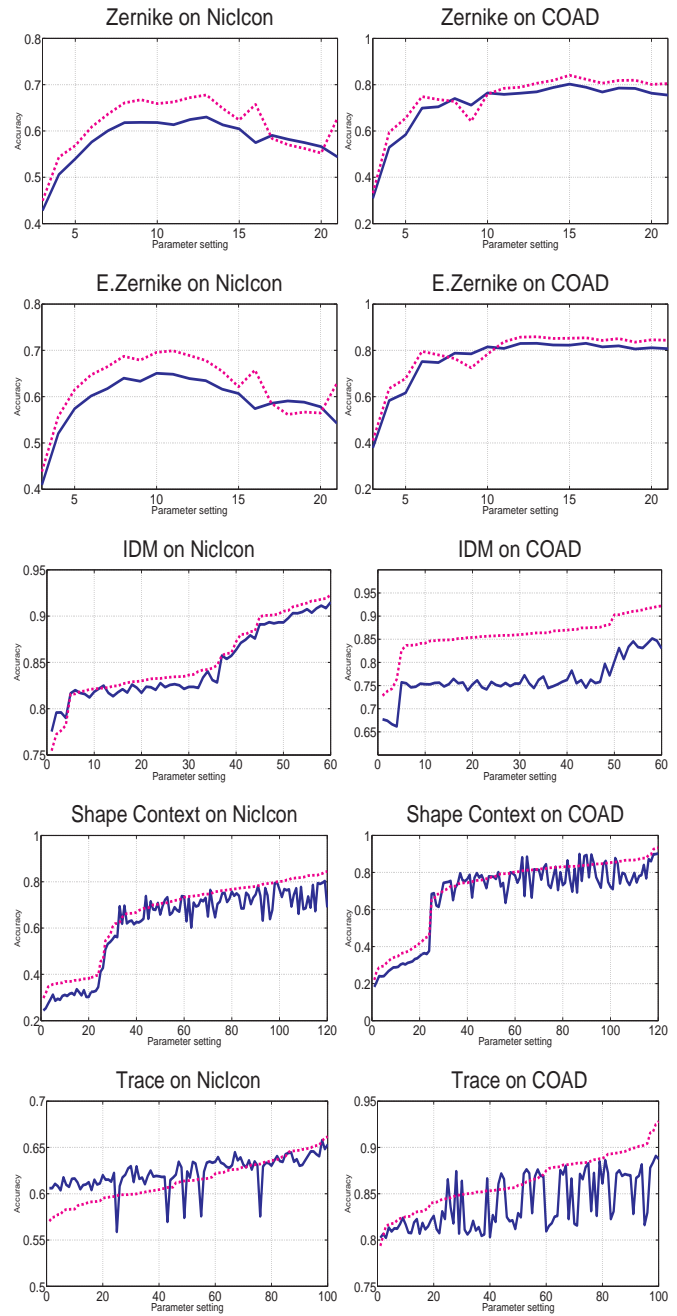

Figure 5: *The mean accuracies for classifier combination methods and individual classifiers. Each plot shows the accuracies of all possible feature subsets for a particular database and feature extraction method sorted by its mean combination accuracy (dashed lines). The horizontal axes show the index of the subset with the ith lowest combined accuracy. The solid lines show the mean of individual classifiers on the same subsets.*

binations show that the differences are statistically significant ($p = 0.05$).

As seen in these figures, classifier combination also results in smoother curves, and reduces the probability of getting low recognition rates when stuck with a particularly unlucky parameter setting.

5.2. Information Fusion

The experiments described above focus on the performance of individual feature extraction methods. One issue that has not been sufficiently explored is whether the information extracted by different feature extraction algorithms are complementary, and whether they can be combined to increase recognition accuracy.

We answer this question within the Maximum Relevance & Minimum Redundancy feature selection framework [PLD05]. By selecting subsets of features generated by all five feature extraction methods, we construct increasingly larger feature vectors, and then look at how many features are contributed by each feature extraction method.

The Maximum Relevance & Minimum Redundancy (MRMR) framework allows one to select the k maximally relevant and minimally redundant features from a total set of K features where $k < K$.

Assume that the respective lengths of feature vectors generated by IDM, ETT, shape contexts, Zernike moments, and extended Zernike moments are l_1, l_2, l_3, l_4, and l_5. So we have a total of l features, where $l = l_1 + l_2 + l_3 + l_4 + l_5$. Now we can use MRMR to generate the k maximally relevant and minimally redundant features from the total set of l features for $k = 1, 2, ..., l$, and measure the classification accuracies.

Fig 6 and Fig 7 show the classification accuracies for each set plotted against the number of features in the set. As seen in these graphs, classification accuracies show a sharp increase as initial features are added. [‡] Past a certain point the accuracy peaks, and starts declining. This is where irrelevant features unavoidably make their way into the top k maximally relevant and minimally redundant features. The peak occurs for $k = 82$ for the COAD database, and $k = 257$ for NicIcon. These values correspond to optimal feature subsets.

It is instructive to look at the number of features contributed to the optimal set by each feature extraction method. Fig 8.a shows the breakdown of the 82 features into the five categories, and as seen here IDM contributes the most number of features, followed by ETT, and shape context. Interestingly, the two Zernike-based features don't contribute any features at all. For the NicIcon database, the ordering of feature extraction methods is the same, but now a few of the Zernike-based features are also found to be useful (Fig. 8.a).

[‡] The accuracy measurements for MRMR were done using 5 fold cross validation.
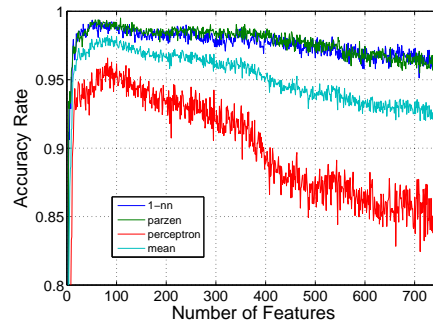

Figure 6: *Classification accuracies for feature sets built incrementally using MRMR, plotted against the number of features in the sets (COAD database). We used the relatively faster Parzen, Perceptron, and 1-nearest-neighbor classifiers to obtain the curves. The mean accuracy is also plotted.*

Figure 7: *Classification accuracies for feature sets built incrementally using MRMR, plotted against the number of features in the sets (NicIcon database). We used the relatively faster Parzen, Perceptron, and 1-nearest-neighbor classifiers to obtain the curves. The mean accuracy is also plotted.*
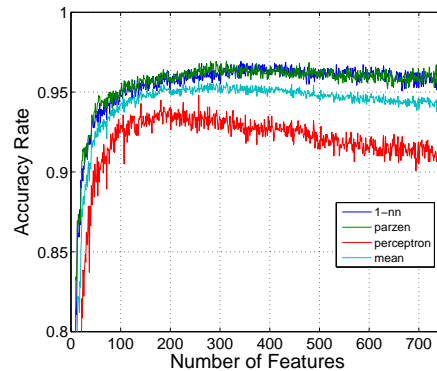
Fig 8.a may give the impression that IDM is overall a superior feature representation, though this is not accurate. The high number of features contributed by IDM is partly due to the fact that IDM features have 720 entries, while ETT and Zernike-based features have 20-30 features. The percentage of features contributed by each feature representation is a more meaningful figure to look at. These values are shown in Fig. 8.b, and as seen here, almost half the features generated by ETT eventually make their way into the optimal combined set for both databases. The percentage of features contributed by IDM remains at modest levels.

Fig 8.a and Fig 8.b illustrate that the feature extraction algorithms studied in this paper are complementary. Furthermore, they illustrate the importance of relevant feature selection in achieving good accuracies, because too many features
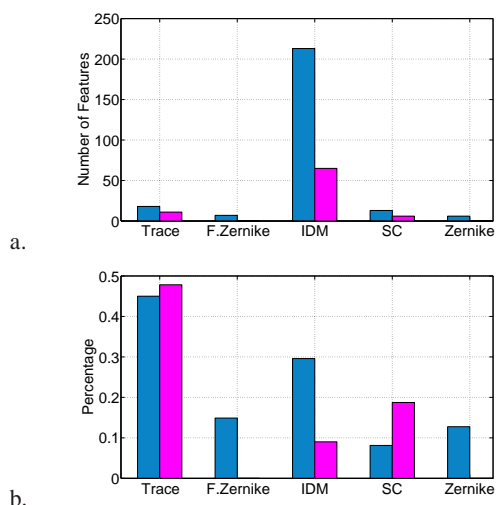
Figure 8: *The distribution of the optimal features selected by MRMR based on the source feature extraction method.*

Dataset	Feature Size	Training Time	Test Time	Training Speed-up	Testing Speed-up
COAD	901	590	7.27	56.41	4.02
COAD	82	10.46	1.81		
NicIcon	1014	550	12.53	17.33	4.20
NicIcon	257	31.73	2.98		

Table 3: *Amount of time taken for training and testing using full feature sets, and optimal feature sets found through MRMR (in seconds). As seen here, reducing the number of features results in 56-17 fold speedups for training, and a 4 fold speedup in recognition.*

may actually hurt recognition performance if they are not all collectively relevant.

It is worth noting that reducing the number of features through MRMR is a costly operation. For example, obtaining the accuracy plots for MRMR plotted in Fig. 6 and Fig. 7 took in the order of 2-3 days for each one of our databases. This is a one-time cost that is paid offline. However, in return, using smaller and optimized features brings a 17-56 fold speedups during training, and a 4 fold speedup during testing (recognition) as seen in Table 3. The offline cost of feature selection can be traded off for the speedups gained in offline training and online recognition depending on the computational resources available to the recognition engine designer, and the end user of the recognition system.

6. Related Work

There are many existing features, and sketch recognition algorithms [SPN02,HD09,MSLL07,SvdP06,GKSS05]. Here, we have focused exclusively on combining image-based fea-

tures because they typically use the three stage "feature extraction, training, classification" framework, and because the features are calculated over fixed image patches, hence result in fixed number of features whose lengths don't vary by the number of strokes.

There are lines of work where the possibility of combining classifiers have been explored. For example, Kara et al. [KS04] combine four image-based classifiers, each of which defines a distance metric between a drawn symbol and learned templates. This is an example of combining multiple classifiers, though the authors use only the "mean-rule" to combine the outputs of the classifiers. Here, we carry the state of art forward by including more classifiers, and more features into our discussion. In addition, we explore the issue of complementarity at the feature level, and show that relevant feature selection is an important issue. Also, we focus on the issue of parameter selection for feature extraction, which has not received much discussion in the literature.

There are also pieces of work where features have been combined in non-image-based frameworks (e.g., constraint-based recognition frameworks). For example, Widmayer et al. [HPW07] and Cheriet et al. [VCC01] describe recognition systems where constraints (as opposed to classifiers or features) based on geometric measurements are combined using various mean-based rules. Anquetil et al. describe a framework that makes it possible to incorporate "statistical recognition" in a constraint multiset grammar based recognition framework [MA09]. The constraint-based and grammar-based frameworks don't fit the usual three stage machine learning pipeline that image-based features use as their common denominator, hence we did not include them in our experiments.

Cates et al. present a sophisticated method for combining representations for sketch recognition [Cat09]. Their work makes use of a rich set of representations including spatial, temporal and conceptual representations. They don't address the feature extraction and classifications issues as we do.

Our work is distinguished from all related work cited above by from several aspects, but most importantly, we specifically focus of feature extraction, address the issue of parameter selection for feature extraction, and we also report new results on feature and classifier combination for boosting performance. We also introduce the ETT feature as a compact feature representation.

7. Future Work

In this paper, we have investigated machine learning related issues that affect the performance of image-based recognition systems. We have excluded constraint-based and grammar-based as well as time-based recognition approaches from our discussion because they do not fit in the three stage machine learning framework as described above.

§ However, it is important to investigate how machine-learning-related design choices affect the performance of these approaches.

For example, more research is needed to understand how missing constraints should be handled in constraint-based recognition frameworks. Or, for example, there is a need to understand the limitations of the Markov assumption and the Naive Bayes assumptions on the performance of recognition accuracy in time-based recognition systems.

8. Summary and Contributions

In this paper we showed that recognition accuracies for image-based recognition systems are closely related to the parameters of the feature extraction methods. Using image-based features effectively requires a careful search for optimal parameter settings. We showed that the search for optimal parameters can be computationally expensive, and that in these cases ensembles of classifiers can be used to improve recognition accuracy for suboptimal feature sets.

We also showed that features computed using different extraction methods are complementary, and combining them within the maximum relevance & minimum redundancy framework is a good way of fusing information provided by different methods and increasing recognition accuracy.

Finally, we introduced the Extended Trace Transform and showed that it can be used for sketch recognition. ETT results in shorter (more compact) feature vectors compared to IDM (30 vs 720 features). Also, when fed to a feature selection algorithm such as MRMR, a larger percentage of ETT features make their way into the final set of features, which confirms that ETT is a compact and highly informative feature representation.

References

[BMP02] BELONGIE S., MALIK J., PUZICHA J.: Shape matching and object recognition using shape contexts. *IEEE T. on Pattern Analysis and Machine Intelligence* (2002), 509–522.

[Cat09] CATES S.: *Combining Representations for Improved Sketch Recognition*. PhD thesis, Massachusetts Institute of Technology, September 2009.

[GKSS05] GENNARI L., KARA L. B., STAHOVICH T. F., SHIMADA K.: Combining geometry and domain knowledge to interpret hand-drawn diagrams. *Computers and Graphics 29*, 4 (2005), 547–562.

[HD09] HAMMOND T. A., DAVIS R.: Recognizing interspersed sketches quickly. In *GI '09: Proceedings of Graphics Interface 2009* (Toronto, Ont., Canada, Canada, 2009), Canadian Information Processing Society, pp. 157–166.

[HN04] HSE H., NEWTON A. R.: Sketched symbol recognition using Zernike moments. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1* (2004), pp. 367–370.

[HPW07] HALL A., POMM C., WIDMAYER P.: A combinatorial approach to multi-domain sketch recognition. In *Proceedings of the 4th Eurographics workshop on Sketch-based interfaces and modeling* (New York, NY, USA, 2007), ACM, pp. 7–14.

[KH90] KHOTANZAD A., HONG Y.: Invariant image recognition by Zernike moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 12*, 5 (1990), 489–497.

[KHDM98] KITTLER J., HATEF M., DUIN R. P. W., MATAS J.: On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell. 20*, 3 (1998), 226–239.

[KP01] KADYROV A., PETROU M.: The trace transform and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2001), 811–828.

[KS04] KARA L., STAHOVICH T.: An image-based trainable symbol recognizer for sketch-based interfaces. *AAAI Fall Symposium Series 2004: Making Pen-Based Interaction Intelligent and Natural* (2004), 501–517.

[LY99] LIU Y., YAO X.: Ensemble learning via negative correlation. *Neural Networks 12*, 10 (1999), 1399–1404.

[MA09] MAC S., ANQUETIL E.: Eager interpretation of on-line hand-drawn structured documents: The dali methodology. *Pattern Recognition 42*, 12 (2009), 3202 – 3214. New Frontiers in Handwriting Recognition.

[Man97] MANUAL F.: 101-5-1, Operational Terms and Graphics. *Washington, DC, Department of the Army 30* (1997).

[MSLL07] MAS J., SANCHEZ G., LLADOS J., LAMIROY B.: An incremental on-line parsing algorithm for recognizing sketching diagrams. In *ICDAR '07: Proc. of the Ninth Int. Conf. on Document Analysis and Recognition* (2007).

[OD09] OUYANG T., DAVIS R.: A visual approach to sketched symbol recognition. In *Proc. International Joint Conferences on Artificial Intelligence* (2009).

[Olt07] OLTMANS M.: *Envisioning Sketch Recognition: A Local Feature Based Approach to Recognizing Informal Sketches*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, May 2007.

[PLD05] PENG H., LONG F., DING C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence 27*, 8 (2005).

[SPN02] SHILMAN M., PASULA H., NEWTON S. R. R.: Statistical visual language models for ink parsing. In *AAAI Spring Symposium on Sketch Understanding* (2002), pp. 126–32.

[SvdP06] SHARON D., VAN DE PANNE M.: Constellation models for sketch recognition. *Eurographics Workshop on Sketch Based Interfaces and Modeling* (2006).

[VCC01] VALOIS J.-P., COTE M., CHERIET M.: Online recognition of sketched electrical diagrams. *Document Analysis and Recognition, International Conference on 0* (2001), 0460.

[Vuu] VUURPIJL L.: The nicicon database. http://unipen.nici.ru.nl/NicIcon/.

[WNvGV09] WILLEMS D., NIELS R., VAN GERVEN M., VUURPIJL L.: Iconic and multi-stroke gesture recognition. *Pattern Recogn. 42*, 12 (2009), 3303–3312.

---

§ Constraint and grammar-based approaches use varying number of constraints and productions to describe objects, hence they are unlike the fixed-length features of image-based approaches. Time-based methods also don't fit into our evaluation framework, because existing approaches to time-based recognition model sketches as sequences of strokes, and in these frameworks the number of observations (features) generated for an object varies as a function of the number of strokes/primitives used while drawing the object.