The UJIpenchars Database: A Pen-Based Database of Isolated Handwritten Characters

D. Llorens*, F. Prat*, A. Marzal*, J. M. Vilar*, M. J. Castro†, J. C. Amengual*, S. Barrachina*, A. Castellanos*, S. España†, J. A. Gómez†, J. Gorbe†, A. Gordo*, V. Palazón*, G. Peris*, R. Ramos-Garijo*, F. Zamora†

*Universitat Jaume I, Castellón, Spain
{dllorens, fprat, amarzal, jvilar, jcamen, barrachi, castella, agordo, palazon, peris, ramosgar}@uji.es

†Universidad Politécnica de Valencia, Spain
{mcastro, sespana, jon, jgorbe, fzamora}@dsic.upv.es

Abstract

The availability of large amounts of data is a fundamental prerequisite for building handwriting recognition systems. Any system needs a test set of labelled samples for measuring its performance along its development and guiding it. Moreover, there are systems that need additional samples for learning the recognition task they have to cope with later, i.e. a training set. Thus, the acquisition and distribution of standard databases has become an important issue in the handwriting recognition research community. Examples of widely used databases in the online domain are UNIPEN, IRONOFF, and Pendigits. This paper describes the current state of our own database, UJIpenchars, whose first version contains online representations of 1 364 isolated handwritten characters produced by 11 writers and is freely available at the UCI Machine Learning Repository. Moreover, we have recently concluded a second acquisition phase, totalling more than 11 000 samples from 60 writers to be made available in short as UJIpenchars2.

1. Introduction

Pen-based input is experiencing great demand since the advent of pen-based computers and devices, such as Tablet PCs, Personal Digital Assistants, digitising tablets, and pressure sensitive screens. For users, a very natural way of pen-based interaction is, simply, handwriting; on the other side, the corresponding device needs an online recogniser accepting such kind of input, i.e. a time-ordered sequence of information about the pen position and, maybe, other data such as its velocity or acceleration (Plamondon and Srihari, 2000). Obviously, the availability of large amounts of data is a fundamental prerequisite for building online handwriting recognition systems. Any system needs a test set of labelled samples for measuring its performance along its development and guiding it. Moreover, there are systems that need additional samples for learning the recognition task they have to cope with later, i.e. a training set. Thus, the acquisition and distribution of standard databases has become an important issue in the handwriting recognition research community. Examples of widely used databases in the online domain are UNIPEN (Guyon et al., 1994), Pendigits (Alpaydın and Alimoğlu, 1998), and IRONOFF (Viard-Gaudin et al., 1999), but not all databases are freely available. The IRONOFF database can be bought from the University of Nantes. Regarding UNIPEN, different data subsets are available in different ways. For instance, at the Products page of the International UNIPEN Foundation (iUF) website[1], references to two such subsets can be found: the UNIPEN collection release #1, which can be bought from the iUF, and the UNIPEN-ICROW-03 benchmark set, freely downloadable. Finally, Pendigits is freely available at the UCI Machine Learning Repository (Asuncion and Newman, 2007).

When our research group began working on handwriting recognition, our first target was to develop a text input panel based on our own template-based recognition engine for isolated characters (Ramos-Garijo et al., 2007; Prat et al., 2007). Thus, we needed an appropriate database for training and testing our engine along its development. Among the free databases we found in the web, the UNIPEN-ICROW-03 benchmark set is not labelled at the character level and Pendigits, though very useful for empirically comparing our engine with others in the literature, does not provide letter samples (only digits), so we decided to collect our own database of isolated handwritten characters.

This paper describes the current state of our database, UJIpenchars, whose first version (Llorens et al., 2007) contains online representations of 1 364 isolated handwritten characters produced by 11 writers and is freely available, as Pendigits is, at the UCI Machine Learning Repository. Moreover, we have recently concluded a second acquisition phase, totalling more than 11 000 samples from 60 writers to be made available in short as UJIpenchars2.

2. UJIpenchars

We created our first character database, UJIpenchars, by collecting samples from 11 writers (see some samples in Figure 1). Each writer contributed with ASCII letters (lower and uppercase), digits, and other characters (letters with Spanish diacritics and punctuation marks) that we have not employed in our experiments and are not included in this database version. Two samples have been collected for each pair writer/character, so the total number of samples in this database version is 1 364:

11 writers × 2 repetitions × (2×26 letters + 10 digits).

[1] The iUF website can be found at `http://unipen.org`.

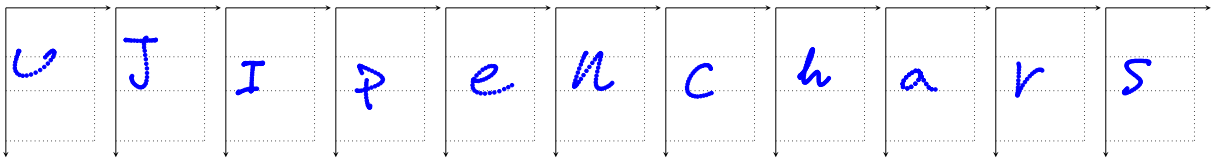


Figure 1: Some samples from UJIpenchars, one for each writer

Moreover, we have defined a standard writer-independent classification task for all database users to evaluate their recognisers in the same conditions.

2.1. The Acquisition

The handwriting samples were collected on a Toshiba Portégé M400 Tablet PC using its cordless stylus. Each of the 11 writers completed 2 non-consecutive sessions. In each session, the corresponding writer was asked to write one exemplar for each character in a fixed set including ASCII letters (lower and uppercase) and digits, along with other characters omitted from this database version (but to be included in UJIpenchars2). The acquisition program shows a set of boxes on the screen, one for each required character, and writers were told to write only inside those boxes (see Figure 2). They were instructed to clear the content of the corresponding box by using an on-screen button (marked as X) and try again whenever they made a mistake or were unhappy with the writing of any character. Subjects were monitored only when writing their first exemplars and every sample considered OK by its writer was accepted.

Only X and Y coordinate information was recorded along the strokes by the acquisition program, without, for instance, velocity, acceleration, pressure level values or timing information. Thus, in multi-stroke samples, no information at all was recorded between strokes; however, in this database version we have included a `.DT 100` line in sample files after each stroke, following the Pendigits database criterion (Alpaydın and Alimoğlu, 1998).

We have observed that runs of consecutive points with identical coordinates were frequently acquired inside strokes; such runs were preserved in this database, so it is up to its users to decide whether to avoid them by an appropriate preprocessing step or not.

We have also observed that UJIpenchars includes a few exemplars with some points lying out of its corresponding acquisition box. As said before, we have not discarded any sample considered OK by its writer.

2.2. The Database Format

The distribution of the UJIpenchars database consists of 12 files: an explanation file (`uji.names`) and one file `UJIpenchars-w`*NN* per writer, where *NN* = `01`, `02`... `11`. Each writer file comprises 124 samples and, for each sample, the following attributes can be found:

- The character it represents.
- The class it belongs to, according to our standard task.
- The sequence of strokes it consists of.

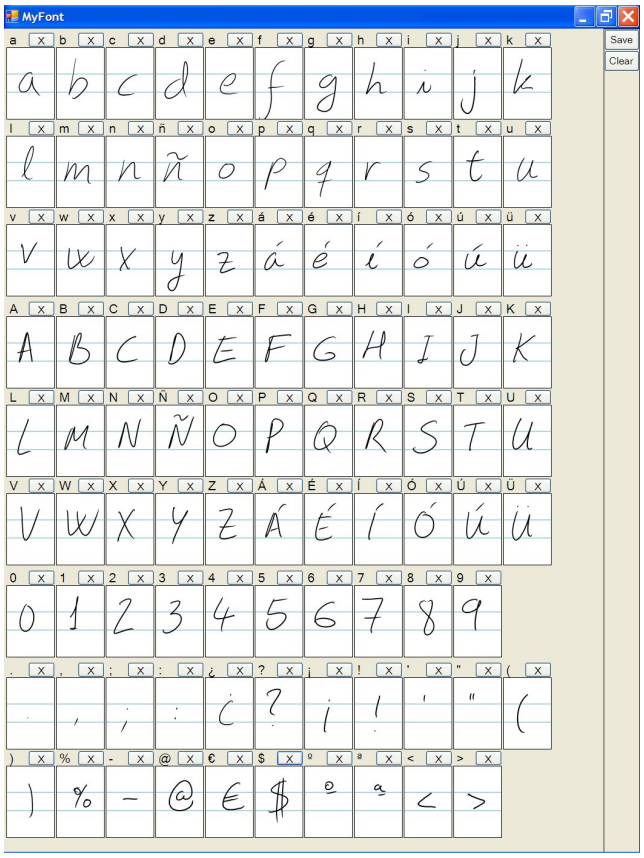


Figure 2: The acquisition program running on the PC

Obviously, when using a sample for testing, only its sequence of strokes should be read to predict its class.

This database is available in a UNIPEN-like format[2], trying to mimic the original Pendigits coding (Alpaydın and Alimoğlu, 1998).

The attributes of each sample appear in its writer file as follows:

- Character name: Each sample begins with a `.SEGMENT` line. The last component of that line shows the character name, one out of 62 possibilities. The complete set of possibilities is shown in the first line of each file, a `.LEXICON` line. The set comprises the ASCII letters from a to z in lower and uppercase, along with the 10 digits.
- Class name: The class name of a sample appears in the `.COMMENT` line that follows its `.SEGMENT` line. This name is one out of 35 possibilities, as explained

[2]See (Guyon, 1994) for a definition of the UNIPEN format.

in Section 2.3. In each file, the complete set of possibilities is shown in `.COMMENT` lines between the `.LEXICON` line and a `.HIERARCHY` one.

- Sequence of strokes: After the `.SEGMENT` and `.COMMENT` lines of a sample, a sequence of one or more strokes follows until the beginning of a new sample or the end of the file. Each stroke begins with a `.PEN_DOWN` line and ends with a sequence `.PEN_UP`, `.DT 100`; in between, a sequence of lines, each one representing X and Y coordinates of a point, where X grows left-to-right and Y grows downwards. Coordinates are integer numbers.

See the Appendix for an example of how a handwritten character is represented in a writer file.

2.3. The Standard Task

The proposed task consists in classifying database samples as belonging to one out of the 35 classes we have defined for this task. Though the number of different characters represented in UJIpenchars is 62, there are only 35 classes because we have not considered a different class for each different character: each one of the 26 letters is considered as a case-independent class, there are 9 additional classes for non-zero digits, and the zero is included in the same class as o's.

Moreover, the proposed task is writer-independent and it consists of 11 leaving-one-writer-out tests, so the effective training set size (for each one of the 1 364 test samples) is 1 240:

10 writers $\times$ 2 repetitions $\times$ (2$\times$26 letters + 10 digits).

2.4. Some Classification Results

On the proposed standard UJIpenchars task, an error rate of 10.9% is achieved with our own real-time k-NN recognition engine (Prat et al., 2007), which is based on approximate Dynamic Time Warping comparisons with prototypes selected by fast, less accurate classification procedures. We have also run experiments with the 1.7 version of the Microsoft Tablet PC SDK recognition engine. These experiments were coded in C# using `Microsoft.Ink.RecognizerContext` objects with an appropriate `WordList` and flags `Coerce` and `WordMode`. The Microsoft recogniser fails the sample category for 14.7% of the database samples.[3]

If we help the Microsoft recogniser by providing it with the dimensions of the acquisition box via its `Guide` property, the error rate drops to 8.4%. These dimensions were not documented in the UJIpenchars distribution made available last year (Llorens et al., 2007), but are to be included in UJIpenchars2.

3. UJIpenchars2

When developing corpus-based recognition engines, getting more data is essential to have the chance to significantly improve a recogniser. Of course, the recogniser performance will not only depend on the amount of data available, but also on finding out how to take the maximum advantage of those data. However, large amounts of data are needed even for studying the best way to use them, so we recently decided to enrich UJIpenchars with a new set of acquisitions in order to keep on improving our own template-based engine. Moreover, we decided to make the resulting database available as UJIpenchars2.

New acquisitions have been carried out at two sites, *Universitat Jaume I* (UJI) and *Universidad Politécnica de Valencia* (UPV), in almost identical conditions. Five new writers have contributed at UJI, where 11 more had already contributed in our first database version. On the other hand, 44 writers have contributed at UPV, totalling 60 writers. Additionally, not only digits and ASCII letters are going to be included in the new database distribution: Spanish vowels with acute accent, the letter "ñ", some punctuation marks (including Spanish ones like "¿" and "¡"), and some symbols will be part of UJIpenchars2 too. The complete set of characters is shown in Figure 2 and Table 1 compares the size and composition of UJIpenchars2 with those of Pendigits and UJIpenchars. The new database, with 11 640 samples, reaches a size similar to Pendigits' and poses a much more complex recognition problem.

At UJI, all samples have been collected using the same hardware and software as described in Section 2.1. On such acquisition platform, each rectangular box shown on the Tablet PC screen measures approximately 13.6$\times$20.4 millimetres (see Figure 3). When the program translates pen positions on the screen into ink coordinates (the ones to be saved on file), it assumes that the origin lies on the top-left corner of the corresponding box, that X grows left-to-right and Y grows downwards, and, at UJI, that there are 100 ink units in each millimetre. At UPV, the same acquisition program has been run on identical hardware, but a different configuration parameter value makes the program translate each millimetre into 152 ink units. Thus, on acquisition files, UPV samples seem to have been collected using larger acquisition boxes. If box homogenisation is needed, it can be easily achieved, for instance, by dividing UPV coordinate values into 1.52.

For UJIpenchars2 distribution, we have decided to fix a division of its set of writers into two disjoint subsets: `(a)` a test set, including none of the 11 UJIpenchars writers, that should only be employed for evaluation in order to fairly compare different recognisers; `(b)` a development set, that can be used for training and validation purposes. Given the new database size, a leaving-one-writer-out strategy does not seem necessary.

Hence, all data for our new database have already been collected and some decisions have been made about how to organise it. After solving a few pending issues, UJIpenchars2 will be freely downloadable from the UCI Machine Learning Repository as a new data set.

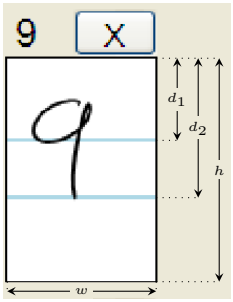
4. Conclusions

The acquisition and distribution of standard databases has become an important issue in the handwriting recognition research community. Database size is a measure of its po-

[3] Thus, the UJIpenchars standard task seems to be much more difficult than Pendigits one: in analogous experiments, our engine achieves an error rate of 0.6% on Pendigits (instead of 10.9%) and Microsoft's obtains a 4.2% (instead of 14.7%).

	Pendigits (44 writers)		UJIpenchars (11 writers)		UJIpenchars2 (60 writers)	
	Characters	Samples	Characters	Samples	Characters	Samples
Digits	10	11 992	10	220	10	1 200
Letters	–	–	52	1 144	66	7 920
ASCII	–	–	52	1 144	52	6 240
Non-ASCII	–	–	–	–	14	1 680
Others	–	–	–	–	21	2 520
ASCII	–	–	–	–	16	1 920
Non-ASCII	–	–	–	–	5	600
Total	10	11 992	62	1 364	97	11 640

Table 1: Size of some free databases of isolated handwritten characters



	Screen units		Ink units	
	Pixels	Millimetres	UJI	UPV
Box width, w	78	$\sim$ 13.6	$\sim$ 1 360	$\sim$ 2 060
Distance d_1	43	$\sim$ 7.5	$\sim$ 750	$\sim$ 1 140
Distance d_2	73	$\sim$ 12.7	$\sim$ 1 270	$\sim$ 1 930
Box height, h	117	$\sim$ 20.4	$\sim$ 2 040	$\sim$ 3 100

Figure 3: An acquisition box and its dimensions

tential usefulness, but not the only one. Accessibility has its own relevance, so when we began working on isolated character recognition and did not find an appropriate database in the web to immediately begin experimenting with, we decided to collect a small one and make it freely available at the same web repository where we had found Pendigits (Alpaydın and Alimoğlu, 1998), a very estimable database but restricted to digit samples. Our small database, including digits and letters, is UJIpenchars (Llorens et al., 2007), currently available at the UCI Machine Learning Repository (Asuncion and Newman, 2007). In this paper, UJIpenchars has been presented in detail, along with the steps we have taken for building a new, larger database to be called UJIpenchars2. All data for this new database have already been collected, totalling more than 11 000 samples (including some non-ASCII Spanish letters and symbols) from 60 writers. In short, UJIpenchars2 will also be freely downloadable from the UCI Machine Learning Repository.

5. Acknowledgements

This work has been partially supported by the Spanish *Ministerio de Educación y Ciencia* (TIN2006-12767 and Consolider Ingenio 2010 CSD2007-00018), the *Generalitat Valenciana* (GV06/302), and the *Fundació Caixa Castelló - Bancaixa* (P1·1B2006-31).

6. References

E. Alpaydın and F. Alimoğlu. 1998. Pen-Based Recognition of Handwritten Digits (original, unnormalized version). Data set available at (Asuncion and Newman, 2007), September.

Arthur Asuncion and David Newman. 2007. UCI Machine Learning Repository. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Isabelle Guyon, Lambert Schomaker, Réjean Plamondon, Mark Liberman, and Stan Janet. 1994. UNIPEN project of on-line data exchange and recognizer benchmarks. In *Proc. of the 12th IAPR International Conference on Pattern Recognition*, pages 29–33, Jerusalem, Israel, October.

Isabelle Guyon. 1994. UNIPEN 1.0 Format Definition. `http://unipen.nici.ru.nl/unipen.def`.

D. Llorens, F. Prat, A. Marzal, and J. M. Vilar. 2007. UJIpenchars: A Pen-Based Classification Task for Isolated Handwritten Characters. Data set available as UJI Pen Characters at (Asuncion and Newman, 2007), June.

Réjean Plamondon and Sargur N. Srihari. 2000. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(1):63–84, January.

Federico Prat, Andrés Marzal, Sergio Martín, and Rafael Ramos-Garijo. 2007. A two-stage template-based recognition engine for on-line handwritten characters. In

Proc. of the Asia-Pacific Workshop 2007 on Visual Information Processing, pages 77–82, Tainan, Taiwan, December.

Rafael Ramos-Garijo, Sergio Martín, Andrés Marzal, Federico Prat, Juan Miguel Vilar, and David Llorens. 2007. An input panel and recognition engine for on-line handwritten text recognition. In Cecilio Angulo and Lluís Godo, editors, *Artificial Intelligence Research and Development*, volume 163 of *Frontiers in Artificial Intelligence and Applications*, pages 223–232. IOS Press.

Christian Viard-Gaudin, Pierre Michel Lallican, Stefan Knerr, and Philippe Binter. 1999. The IRESTE On/Off (IRONOFF) dual handwriting database. In *Proc. of the Fifth International Conference on Document Analysis and Recognition*, pages 455–458, Bangalore, India, September.

Appendix: A Listing

In order to illustrate how handwritten characters are represented in UJIpenchars files, using UNIPEN format, a fragment of `UJIpenchars-w11` (from line 5287 to 5343) follows:

```
.SEGMENT CHARACTER 85-86 ? "i"
.COMMENT Class [I]

.PEN_DOWN
 502  839
 502  830
 506  813
 508  800
 512  789
 512  789
 515  776
 515  776
 515  776
 518  765
 518  765
 510  780
 502  794
 496  807
 485  828
 477  851
 465  875
 455  903
 443  929
 432  956
 421  987
 410 1016
 399 1048
 389 1078
 381 1114
 377 1145
 376 1176
 379 1204
 386 1227
 399 1245
 417 1258
 437 1268
 463 1267
 486 1265
 512 1250
 543 1226
 570 1200
 598 1166
.PEN_UP
.DT 100

.PEN_DOWN
 688  383
 688  383
 688  383
 692  374
 692  374
 692  374
 708  370
 708  370
 719  381
.PEN_UP
.DT 100
```

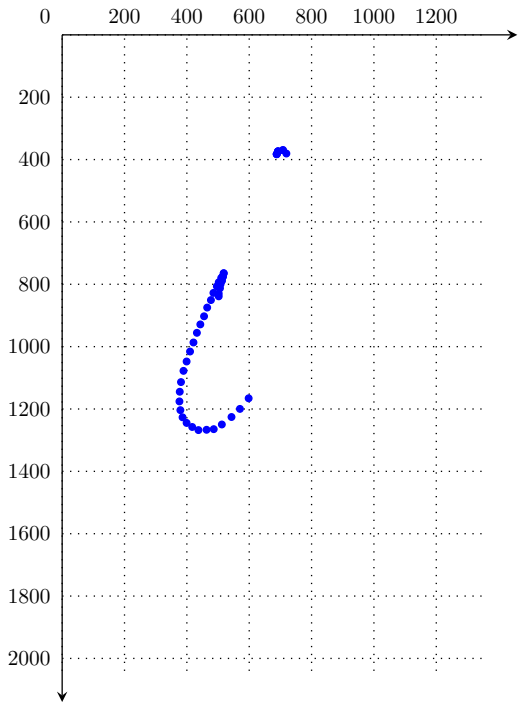


Figure 4: The second lowercase i from writer `w11`

This fragment corresponds to the second lowercase i from writer `w11`, shown in Figure 4, which consists of two strokes numbered as 87 and 88 in the file[4]. Comment in line 5288 tells that the sample belongs to class `[I]`, which has been defined in a previous comment line as including both lower and uppercase i:

```
.COMMENT Class #09: [I] = { "i" , "I" }
```

[4]Strokes are numbered independently in each file, starting from 0.