ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.29, No-1-2, 2021

Word Recognition using Embedded Prototype Subspace Classifiers on a new Imbalanced Dataset

Anders Hast and Ekta Vats
Department of Information Technology
Uppsala University
SE-751 05 Uppsala, Sweden
anders.hast@it.uu.se; ekta.vats@it.uu.se

ABSTRACT

This paper presents an approach towards word recognition based on embedded prototype subspace classification. The purpose of this paper is three-fold. Firstly, a new dataset for word recognition is presented, which is extracted from the Esposalles database consisting of the Barcelona cathedral marriage records. Secondly, different clustering techniques are evaluated for Embedded Prototype Subspace Classifiers. The dataset, containing 30 different classes of words is heavily imbalanced, and some word classes are very similar, which renders the classification task rather challenging. For ease of use, no stratified sampling is done in advance, and the impact of different data splits is evaluated for different clustering techniques. It will be demonstrated that the original clustering technique based on scaling the bandwidth has to be adjusted for this new dataset. Thirdly, an algorithm is therefore proposed that finds k clusters, striving to obtain a certain amount of feature points in each cluster, rather than finding some clusters based on scaling the Silverman's rule of thumb. Furthermore, Self Organising Maps are also evaluated as both a clustering and embedding technique.

Keywords

Subspaces, Embedded Prototypes, Clustering, Deep Learning, Self Organising Maps, t-SNE, Data splits.

1 INTRODUCTION

Recently, *Embedded Prototype Subspace Classification* (EPSC) [HLV19, HL20] has proven to be able to classify datasets containing single digits, characters and even objects, such as the MNIST dataset of handwritten digits [LCB10], E-MNIST containing letters [CATvS17], the Kuzushiji-MNIST dataset containing Japanese handwritten characters [CBK*18], and the Fashion MNIST (F-MNIST) [XRV17] containing small images of clothes and accessories.

The advantage of EPSC compared to deep learning based methods [Sha18] for handwritten text recognition [KDJ18, DKMJ18, SF16] is that EPSC do not require powerful GPU resources in the training process and have no hidden layers, which makes it compact and fast. In general, EPSC learns from the embedding of feature vectors and creates a so-called subspaces from each cluster [KLR*77]. Even though EPSC does not always outperform the state-of-the-art deep learning approaches, it performs significantly as an alternative, where the learning and classification processes are both easy to interpret [Kri19, CPC19], explain [ADRS*19, GSC*19, CPC19], and visualise.

The main contributions of this paper are as follows. First of all a new dataset based on the Esposalles dataset [RFS*13] is presented, where 30 different words were extracted from the given training set. This new dataset can be used for the purpose of evaluating word recognition methods, rather than performing character or digit level recognition. This dataset is by intention heavily imbalanced, which makes it more interesting for real-world problems. Secondly, we present and compare three different methods for computing clusters, aimed at handling imbalanced datasets. Thirdly, an algorithm that finds k seed points for K-means clustering [HW79] is proposed .

2 BACKGROUND

Subspaces have been used for classification in pattern recognition since it was first proposed by Watanabe et al. [WP73] in 1967, and later further developed by Kohonen and others [WLK*67, KLR*77, KO76, KRMV76, OK88]. In general, by computing the norm of the projected feature vector to be classified into each subspace, the process can be regarded as a two layer neural network [OK88, Laa07], where the weights are mathematically defined through Principal Component

DOI:10.24132/JWSCG.2021.29.5

Analysis (PCA) [Laa07]. Another important advantage is that the learning process can easily be visualised, which makes it easy to understand, interpret and explain, as compared to most of the state-of-the-art deep learning approaches.

2.1 Subspace Classification

Herein, we have used the same kind of subspaces that were presented in [HLV19, Laa07, OK88]. Hence, every image to be classified is represented by a feature vector $\mathbf{x}$ with m real-valued elements $\mathbf{x}_j = \{x_1, z_2 ... x_m\}, \in \mathbb{R}$, such that the operations take place in a m-dimensional vector space $\mathbb{R}^m$. Any set of n linearly independent basis vectors $\{\mathbf{u}_1, \mathbf{u}_2, ... \mathbf{u}_n\}$, where $\mathbf{u}_i = \{w_{1,j}, w_{2,j} ... w_{m,j}\}, w_{i,j} \in \mathbb{R}$, which can be combined into an $m \times n$ matrix $\mathbf{U} \in \mathbb{R}^{m \times n}$, span a subspace $\mathscr{L}_\mathbf{U}$

$$\mathscr{L}_\mathbf{U} = \{\mathbf{x} | \mathbf{x} = \sum_{i=1}^{n} \rho_i \mathbf{u}_i, \rho_i \in \mathbb{R}\} \tag{1}$$

where,

$$\rho_i = \mathbf{x}^T \mathbf{u}_i = \sum_{j=1}^{m} x_j w_{i,j} \tag{2}$$

Classification of a feature vector can be performed by projecting $\mathbf{x}$ onto each and every subspace $\mathscr{L}_{\mathbf{U}_k}$. The vector $\hat{\mathbf{x}}$ will in this way be a reconstruction of $\mathbf{x}$, using all vectors in the subspace through

$$\hat{\mathbf{x}} = \sum_{i=1}^{n} (\mathbf{x}^T \mathbf{u}_i) \mathbf{u}_i \tag{3}$$

$$= \sum_{i=1}^{n} \rho_i \mathbf{u}_i \tag{4}$$

$$= \mathbf{U}^T \mathbf{U} x^T \tag{5}$$

By normalising all the vectors in $\mathbf{U}$, the norm of the projected vector can be simplified as

$$||\hat{\mathbf{x}}||^2 = (\mathbf{U}x^T) \cdot (\mathbf{U}x^T) \tag{6}$$

$$= (\mathbf{U}x^T)^2 \tag{7}$$

$$= \sum_{i=1}^{n} {\rho_i}^2 \tag{8}$$

In this way, the feature vector $\mathbf{x}$, which is most similar to the feature vectors that were used to construct the subspace in question $\mathscr{L}_{\mathbf{U}_k}$, will subsequently also have the largest norm $||\hat{\mathbf{x}}||^2$.

2.2 Embedding and Clustering

In general, some group of prototypes are selected for the construction of each subspace by searching for the k nearest neighbors in the feature space, which is a rather time consuming process. The idea of EPSC [HLV19] is on the other hand to use t-distributed stochastic neighbour embedding (t-SNE) [MH08], which is both a visualisation technique as well as a machine learning technique, used to reduce the number of dimensions of high dimensional data (e.g. 2 dimensions in this case). In this process, clusters are formed since t-SNE strives to move similar features (represented by their projected points) closer to each other and dissimilar points further away from each other. Nevertheless, any embedding technique could be used for this purpose, such as Uniform Manifold Approximation and Projection (UMAP) [MH18].

Hast et al. [HLV19] used kernel density estimation (KDE) [CHTT96] and watershed transform on the inverse image to find clusters in a two-dimensional image space. Alternatively, Mean-Shift [CM02, FH75] could be used that also finds the number of clusters depending on the size of the Gaussian Kernel chosen. However, as will be investigated further herein, other algorithms that require specifying the exact number of clusters, such as K-means [HW79] could also be used. Nevertheless, depending upon the problem at hand, other similar algorithms such as DBSCAN [EKSX96] can also be employed.

It is also studied that the clustering techniques that work well for balanced dataset, such as MNIST [LCB10], does not perform well for imbalanced data, where there is an imbalance in the frequency of occurrence of labels in the dataset. Therefore, this work does not rely on automatic cluster selection based on Silverman's rule of thumb for computing the bandwidth h of the clustering

$$h = \left(\frac{4\sigma^5}{3n} \right)^{1/5} \tag{9}$$

where σ is the standard deviation of n samples.

3 THE PROPOSED APPROACH

Instead of using the bandwidth for clustering, better performance was achieved by computing k clusters, striving for these clusters to contain a certain predefined number of features n_f. An intuitive choice is therefore to use K-means clustering approach. Experimentally, it was found that $n_f = 40$ was by large an optimal choice for the number of clusters. However, typically this value depends on the data, and can therefore be evaluated in the learning process with respect to the type of recognition task at hand.

A drawback of K-means clustering is that it finds k clusters by initialising k random seed points, and is not confident if the same clusters will be found when the algorithm is executed multiple times. Since repeatability is important, a deterministic approach was devised that makes use of Kernel Density Estimation (KDE). In general, a certain value of σ is used to splat Gaussians on a

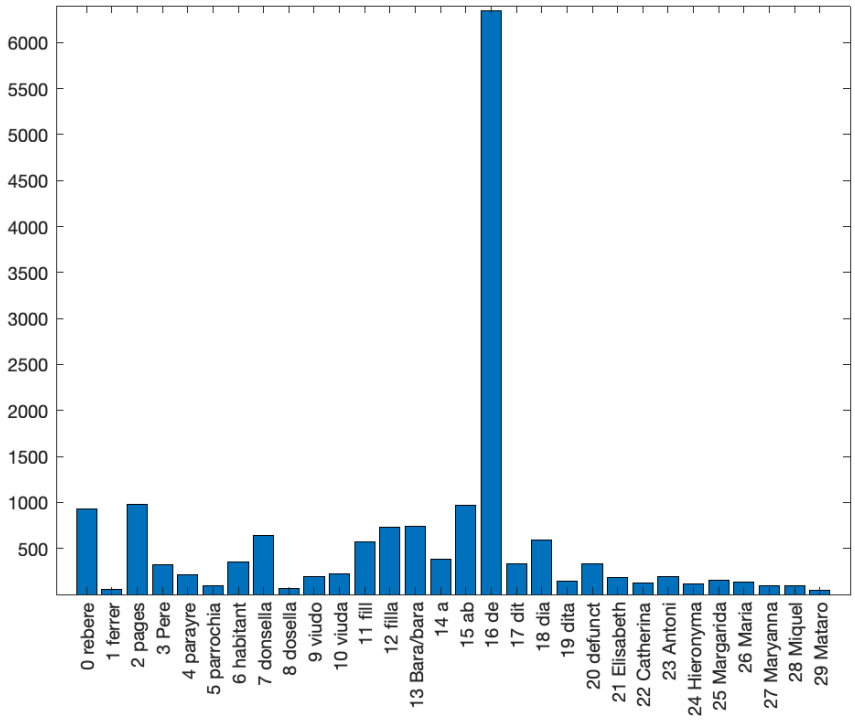


Figure 1: Distribution of words in the "imbalanced" dataset.

small image. The cluster centres are estimated by non-maximum suppression. Depending on the number of maximum (clusters) found, the σ is adjusted, and the process is repeated until k cluster centers are at hand. These are subsequently used to initialise the seed points for K-means, in order to make the algorithm find clusters centered around those maximum points.

The same adaptive procedure can also be used for the original idea of KDE and watershed to obtain k clusters, and are therefore referred to as "adaptive" in our experiments. The original approach was used in the experiments so that the improvement by the adaptive approaches could also be evaluated.

Furthermore, a self-organizing map (SOM) [Koh82] can be used, which is a dimensionality reduction technique based on unsupervised competitive learning of an artificial neural network (ANN). It generates a 2D map representation of an input space of the training samples, where the features are placed in buckets. They also demonstrate well for finding k clusters, with an advantage of having the embedding itself as part of the clustering process. However, the disadvantage is that SOM cannot be used to produce elegant scatter plots like t-SNE, as the points end up in the buckets. Nonetheless, an improved version of SOM can be used to visualise the scatter plots, called as EmbeddSOM [KKV19], which is based on FlowSOM [VGCVH*15].

SOM was configured in a way that it would strive to use a $n \times m$ map, where $n \times m = k$. However, since it is rather slow and impractical to be used for large classes,

Figure 2: The placement of each word in the 90×160 rectangular bounding box. The background is removed but the word is not binarised.

an upper limit of $n = m = 6$ was set. Similarly, an upper limit for K-means was set to a maximum of $k = 40$. All of these limits were set ad-hoc and can be changed depending on the dataset at hand. Nonetheless, all the three approaches are therefore referred to as "Adaptive" since they adaptively set the number of wanted clusters depending on the size of each class.

4 INTRODUCING THE NEW DATASET

To the best of authors knowledge and taking inspiration from the MNIST dataset, this is the first attempt at creating a dataset for handwritten word recognition for a public research domain, which is based on the Esposalles database [RFS*13]. In order to render the dataset more significant and challenging, 30 words were chosen, producing a total of 16354 word images, where some words are very similar in nature, and Figure 1

Figure 3: The 51 occurrences of the word "ferrer", with highlights on different handwriting styles. It can be noted how the character "f" is written in different styles.

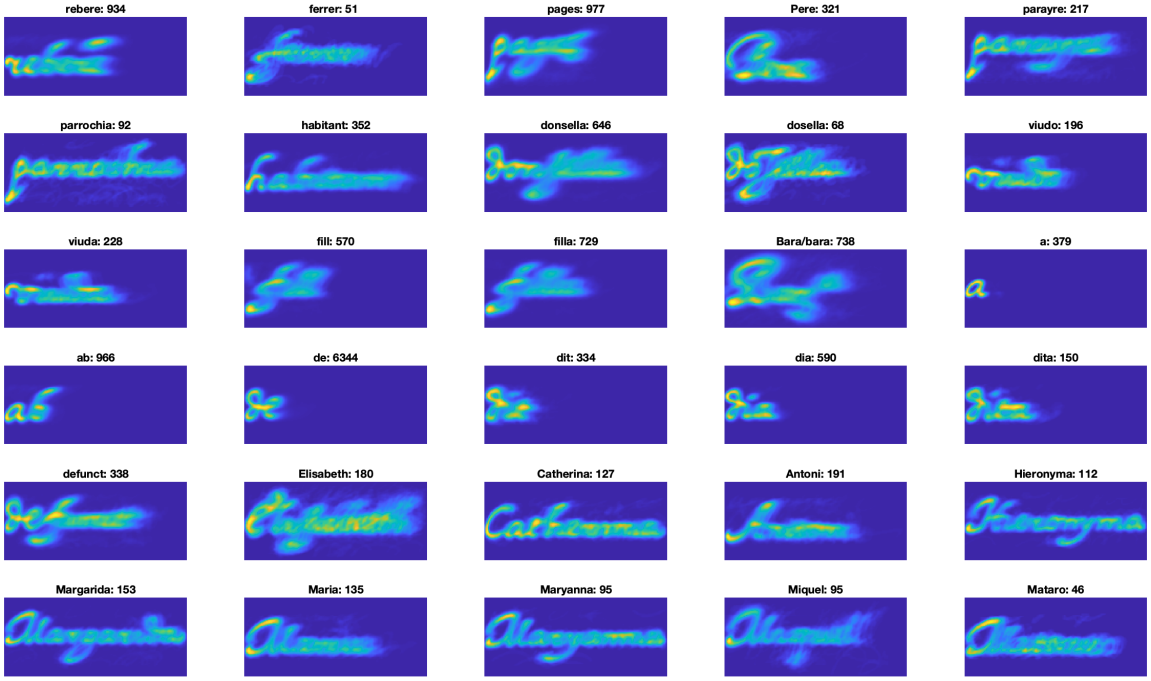


Figure 4: Heatmaps of all the words in the dataset, clustered by the respective class. The number of images per class are shown above each heatmap. Figure best viewed in color.

presents the distribution of words in the dataset. As can be seen, the dataset is heavily imbalanced, and the Shannon equitability index is only 0.7444, as compared to the balanced MNIST dataset with a Shannon equitability index of 0.9994.

Each word has been extracted using the bounding box coordinates provided as part of the Esposalles database. However, the input image is processed as follows. To begin with, background removal using [VHS17] is performed on each page, and the word is extracted.

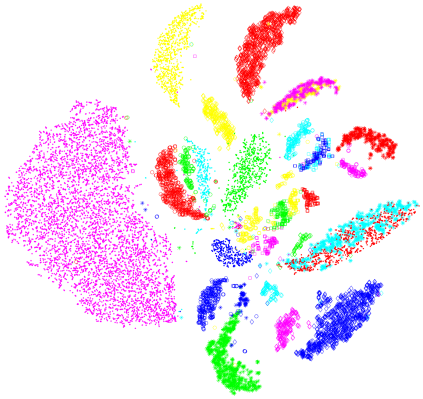
There exist some noise in the form of small blobs due to bleed through which is removed automatically, and the rectangular bounding boxes are adjusted so they perfectly encapsulate the word region [VH17]. Finally, the word is centered (or normalised), and placed to the left in a 90×160 rectangular box to accommodate words with larger length (e.g. as long as 9 characters), as shown in Figure 2. This also means that the word will be placed a little bit differently, with respect to its core, depending on whether the word has ascenders and/or descenders.

The variability within a certain class can be quite large depending on the number of writers, and whether there exists more noise in the image. The variation can be visualised as heatmaps of each class, as presented in Figure 4. These are created by adding all words belonging to a class into one single image, and the resulting values are colour coded. It can be noted that some words demonstrate a larger variation than others, as the heatmap is visually more blurry.

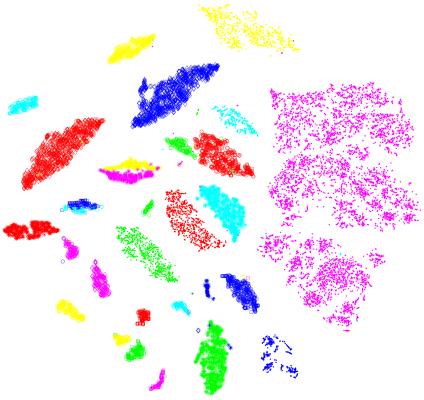
For ease of use, no stratified sampling is done in advance, or in other words, the word images are not split into predefined training and testing sets. Such splits are commonly provided for MNIST and many other popular datasets, where some also provide a validation set. However, this work allows the researchers with the flexibility to split the data in their preferred way and evaluate different properties of the dataset as well as the machine learning algorithm.

The dataset can be visualised using t-SNE or any other embedding technique, such as UMAP [MH18]. In Figure 5, t-SNE was used on the following: 5a: the word images, 5b: Histogram of Gradients (HOG) feature vectors [DT05], and 5c: mFFT, which are Fast Fourier Transform (FFT) based features with combinations of some of the most significant elements of the magnitude of the FFT.

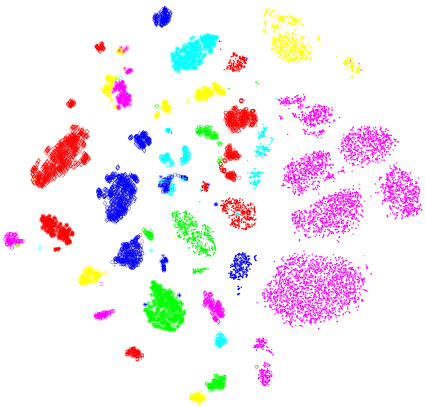
In Figure 5a, a big magenta coloured cluster can be observed where each point representing the word "de" is split into smaller clusters depending on its characteristic look in both Figure 5b and Figure 5c. Moreover, the cluster to the middle right containing the two similar words, "viudo" and "viuda", is mixed in Figure 5a, while in Figure 5b and Figure 5c it is automatically split into two distinct clusters. This suggests using efficient feature vectors instead of using simply the word images. For simplicity, hand crafted features are used, but depending upon the problem at hand and the availability of the resources, CNN based features can also be used.



(a) Word images used as features.



(b) HOG features.



(c) mFFT features.

Figure 5: Visualisation of how effectively different features can separate different classes. Figure best viewed in color.

5 RESULTS

Since the dataset is heavily imbalanced, the so-called Macro Average Arithmetic (MAA) [AAVPS13] is computed instead of the overall accuracy. The latter can be rather misleading for imbalanced datasets, when for instance a few classes with many occurrences and high accuracy can skew the overall accuracy. Figure 6 highlights that most classes particularly those with many occurrences perform well, while some with just a few occurrences yield a low accuracy. Hence, it is a better approach to compute the accuracy of each class indi-

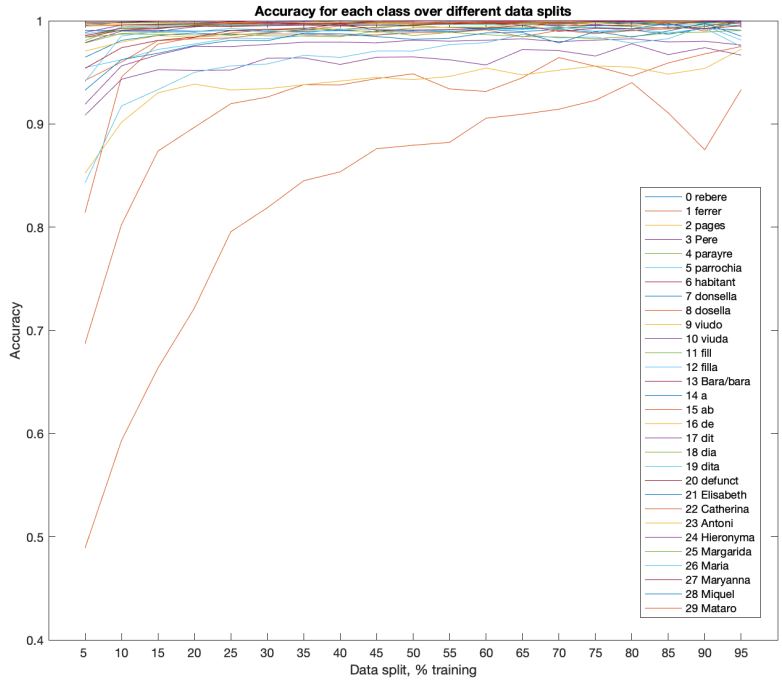


Figure 6: Accuracy computed for each class individually. Classes with many occurrences generally have higher accuracy than classes with fewer occurrences.

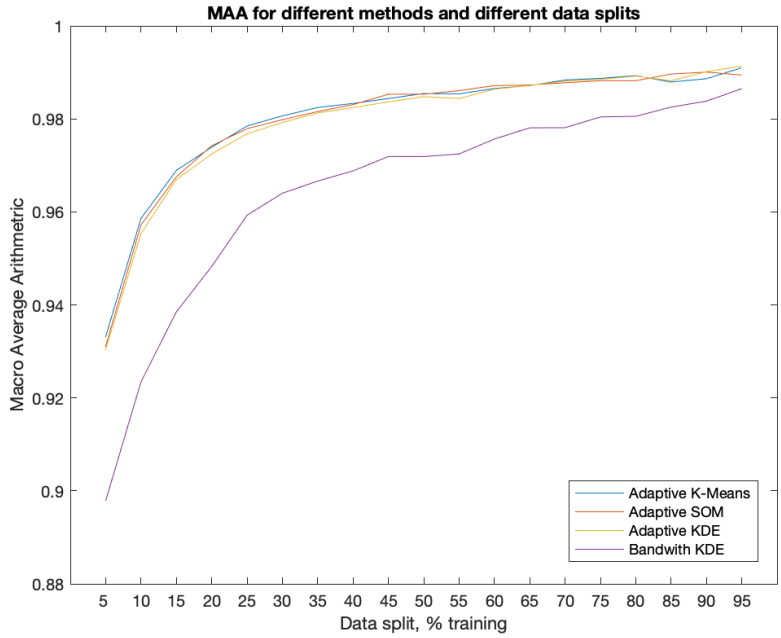


Figure 7: Macro Average Arithmetic (MAA) for different methods and different data splits.

vidually, and then compute the average. The accuracy of each class is

$$ACC_j = \frac{CC_j}{N_j} \tag{10}$$

where CC_j are the number of correctly classified words in class j and N_j is the number of samples (i.e. words) in the same class.

The MAA is defined as the arithmetic average of the partial accuracies of each class

$$MAA = \frac{\sum_{i=1}^{J} ACC_i}{J} \tag{11}$$

where J is the number of classes.

Figure 7 presents a comparison with the original method, referred to as "bandwidth KDE", since it is

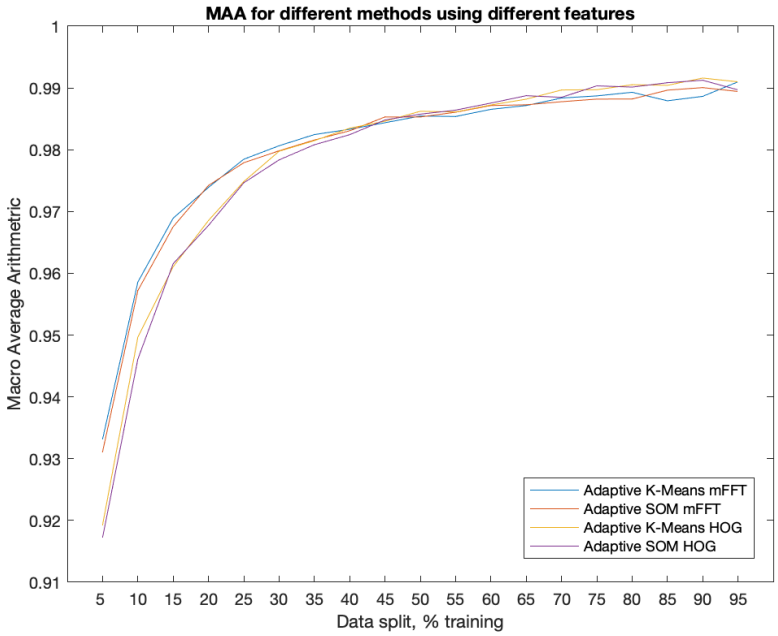


Figure 8: Macro Average Arithmetic (MAA) for different methods and different feature vectors and data splits.

based on scaling the Silverman's rule of thumb and generating clusters based on the distribution of points in 2D space. This will yield clusters with rather varying number of points. The other three approaches (SOM, K-Means and KDE) are based on the aforementioned idea of obtaining the clusters that have a similar number of points in them. As can be seen, using this idea on SOM, K-Means or KDE, all behave quite similar, and perform better than bandwidth KDE. For the proposed dataset and the setting of parameters, it is hard to choose a winner. However, it should be noted that the whole parameter space was not spanned to find the best settings, since it also depends on the dataset at hand. Hence, this is proposed for future research. All experiments were repeated 40 times for each data split and the average result is reported in the plots. This also highlights that the learning set and the testing set in each run contained different permutations of words. The same random seed was used for each group of experiments depicted by one curve, so that the curves could be compared individually on the same basis.

In Figure 7, the mFFT was used as feature vectors, which has a length of 1116, which is only 7.8% of the original size of the word images ($90 \times 160 = 14400$). It is interesting to see how the EPSC performs with different feature vectors e.g. the HOG, which is 6940 long, i.e. 47.5% of the original images. The results are presented in Figure 8, and it can be noted that the X-axis is different here in order to better observe the difference between the plots. Nonetheless, HOG seems to perform well after around a 50% split, while mFFT is more effective for less learning data, which is encouraging since they are much shorter.

6 DISCUSSION

It is observed that the EPSC handles imbalanced datasets very well, since high accuracy is obtained for most of the classes with just a few occurrences. Of course, just one dataset is not enough to provide an absolute answer. However, the proposed dataset is indeed useful and challenging, with some words classes very similar in nature. EPSC handles the imbalance in the following way. When there are several occurrences in one class, it creates subspaces that capture the variation within that class. Hence, it correctly classifies that class, and the images not belonging to the class will instead be correctly classified by subspaces belonging to other classes. That is, if it has enough learning examples to create the subspaces for those classes. Hence, having several occurrences for one class does not seem to pose big problem. Having too few, on the other hand, becomes challenging since the subspaces created do not capture the variation very well. For instance only one subspace is created for the word "ferrer" when 5% is used for learning, and that subspace is created from only 3 occurrences. Typically, the word classes that are very similar will suffer more from having a very small variation of word examples to create the subspaces from.

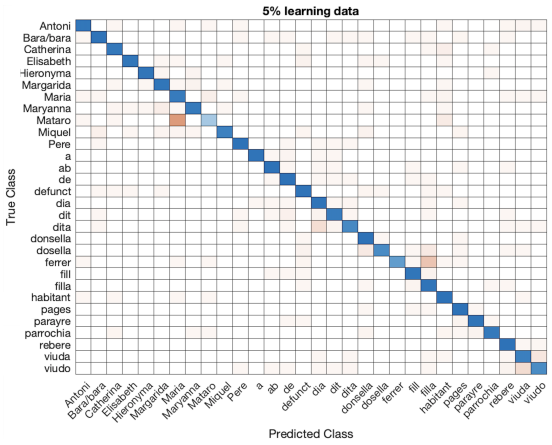
When dealing with imbalanced datasets, one can choose different strategies, such as to over-sample the minority class, under-sample the majority class or generate synthetic samples. When transcribing a document, one can use any of these strategies to make a learning model using the words already transcribed, and perform automatic transcription for the rest. How-

ever, in this paper the main focus was to investigate how well EPSC can handle imbalanced sets and which clustering approaches can be used efficiently. If under-sampling and over-sampling are not suitable strategies for EPSC (since it is desirable to keep as much variation as possible), then data augmentation of the minority classes can potentially be a solution to improve the performance. This is proposed for future research.

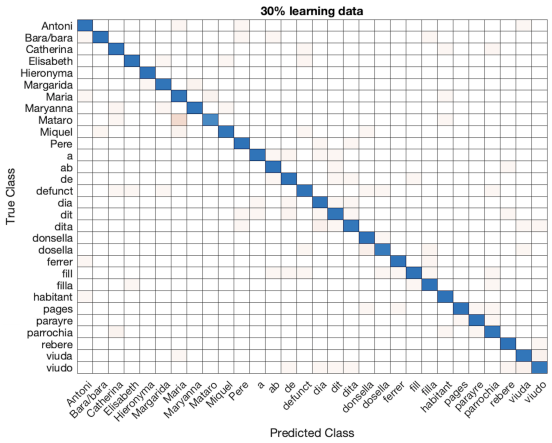
The MAA for all clustering techniques and feature vectors, reach about 98% for a data split of 30% used for training, which can be regarded as a good result for such a challenging dataset. Even for the human eye, it is at times difficult to differ handwritten words like "viudo" from "viuda". It is observed that 17 of the classes have an accuracy over 99.0% for the same data split, 14 lies over 99.5% and 7 are over 99.7%, when using K-means and mFFT. The Confusion matrices for 5%, 30%m and 60% learning data are shown in Figure 9. It can be noted that the word "Mataro" is often misclassified as "Maria", while the words "viudo" and "viuda" are interestingly much less confused.

7 CONCLUSION

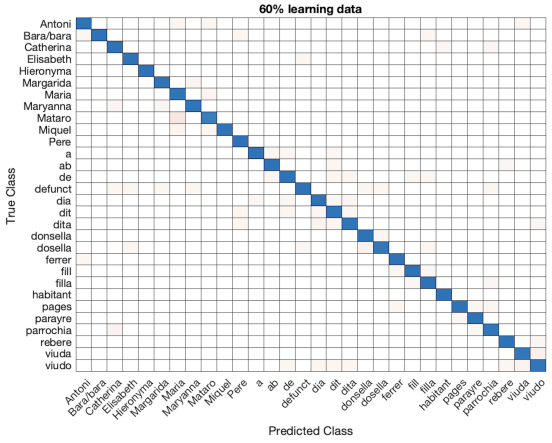
The original approach for clustering for EPSC works very well for balanced datasets. However, imbalanced datasets are generally harder to handle and three different clustering approaches for obtaining subspaces were presented and analysed in this work. Experimental results validate the performance, and all the three approaches performed significantly regardless of the feature vector being used. It did not make a big difference whether t-SNE or SOM were used as embeddings. Neither did, using the proposed adaptive versions of the original idea of clustering (using the watershed transform or K-means), add a significant difference. Hence, all three methods are good candidates for future experiments. However, SOM has the drawback of not being a visualisation technique by its own. All parameters in the EPSC were not systematically investigated for the new heavily imbalanced dataset in the current experiments. Nevertheless, a baseline was given for future experimenting using EPSC. Furthermore, handcrafted features were used as they are fast and simple, rendering the whole pipeline with EPSC fast and simple too. As future work, CNN based features would be evaluated for the given dataset. However, full pipelines of deep learning approaches also need to handle the imbalance in specific ways. Therefore, it will be an interesting area of research for further investigations on using the dataset for different data splits and machine learning algorithms.



(a) Confusion matrix with 5% learning data.



(b) Confusion matrix with 30% learning data.



(c) Confusion matrix with 60% learning data.

Figure 9: Confusion matrices for 5%, 30% and 60% learning data respectively.

8 ACKNOWLEDGMENTS

This work has been partially supported by the Riksbankens Jubileumsfond (Dnr NHS14-2068:1). The computations were performed on resources provided by SNIC through UPPMAX under project SNIC 2020/15-177. The authors wish to thank Raphaela Heil for fruitful discussions in the development of the ideas presented. The presented dataset is publicly available at `https://andershast.com/datasets/`

9 REFERENCES

[AAVPS13] Alejo R., Antonio J. A., Valdovinos R. M., Pacheco-Sánchez J. H.: Assessments metrics for multi-class imbalance learning: A preliminary study. In *Pattern Recognition* (Berlin, Heidelberg, 2013), Carrasco-Ochoa J. A., Martínez-Trinidad J. F., Rodríguez J. S., di Baja G. S., (Eds.), Springer Berlin Heidelberg, pp. 335–343.

[ADRS*19] Arrieta A. B., D'iaz-Rodr'iguez N., Ser J. D., Bennetot A., Tabik S., Barbado A., Garc'ia S., Gil-L'opez S., Molina D., Benjamins R., Chatila R., Herrera F.: Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *ArXiv abs/1910.10045* (2019).

[CATvS17] Cohen G., Afshar S., Tapson J., van Schaik A.: EMNIST: an extension of MNIST to handwritten letters. *CoRR abs/1702.05373* (2017).

[CBK*18] Clanuwat T., Bober-Irizar M., Kitamoto A., Lamb A., Yamamoto K., Ha D.: Deep learning for classical japanese literature. *CoRR abs/1812.01718* (2018).

[CHTT96] Carbon M., Hallin M., Tat Tran L.: Kernel density estimation for random fields: the l 1 theory. *Journal of nonparametric Statistics 6*, 2-3 (1996), 157–170.

[CM02] Comaniciu D., Meer P.: Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 5 (May 2002), 603–619.

[CPC19] Carvalho D. V., Pereira E. M., Cardoso J. S.: Machine learning interpretability: A survey on methods and metrics. *Electronics 8*, 8 (Jul 2019), 832.

[DKMJ18] Dutta K., Krishnan P., Mathew M., Jawahar C.: Improving cnn-rnn hybrid networks for handwriting recognition. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)* (2018), IEEE, pp. 80–85.

[DT05] Dalal N., Triggs B.: Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (June 2005), vol. 1, pp. 886–893 vol. 1.

[EKSX96] Ester M., Kriegel H.-P., Sander J., Xu X.: A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (1996), KDD96, AAAI Press, pp. 226–231.

[FH75] Fukunaga K., Hostetler L.: The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory 21*, 1 (January 1975), 32–40.

[GSC*19] Gunning D., Stefik M., Choi J., Miller T., Stumpf S., Yang G.-Z.: Xai—explainable artificial intelligence. *Science Robotics 4*, 37 (2019).

[HL20] Hast A., Lind M.: Ensembles and cascading of embedded prototype subspace classifiers. *Journal of WSCG 28*, 1/2 (2020), 89–95.

[HLV19] Hast A., Lind M., Vats E.: Embedded prototype subspace classification : A subspace learning framework. In *The 18th International Conference on Computer Analysis of Images and Patterns (CAIP)* (2019), Lecture Notes in Computer Science, pp. 581–592.

[HW79] Hartigan J. A., Wong M. A.: Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics) 28*, 1 (1979), 100–108.

[KDJ18] Krishnan P., Dutta K., Jawahar C.: Word spotting and recognition using deep embedding. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)* (2018), IEEE, pp. 1–6.

[KKV19] Kratochvíl M., Koladiya A., Vondrášek J.: Generalized embedsom on quadtree-structured self-organizing maps. *F1000Research* (2019).

[KLR*77] Kohonen T., Lehtiö P., Rovamo J., Hyvärinen J., Bry K., Vainio L.: A principle of neural associative memory. *Neuroscience 2*, 6 (1977), 1065 – 1076.

[KO76] Kohonen T., Oja E.: Fast adaptive formation of orthogonalizing filters and associative memory in recurrent networks of neuron-like elements. *Biological Cybernetics 21*, 2 (Jun 1976), 85–95.

[Koh82] Kohonen T.: Self-organized formation of topologically correct feature maps. *Biological Cybernetics 43*, 1 (Jan. 1982), 59–69.

[Kri19] Krishnan M.: Against interpretability: a critical examination of the interpretability problem in machine learning. *Philosophy & Technology* (2019).

[KRMV76] Kohonen T., Reuhkala E., Mäkisara K., Vainio L.: Associative recall of images. *Biological Cybernetics 22*, 3 (Sep 1976), 159–168.

[Laa07] Laaksonen J.: *Subspace classifiers in recognition of handwritten digits*. G4 monografiaväitöskirja, Helsinki University of Technology, 1997-05-07.

[LCB10] LeCun Y., Cortes C., Burges C.: Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist 2* (2010).

[MH08] Maaten L. v. d., Hinton G.: Visualizing data using t-sne. *Journal of machine learning research 9*, Nov (2008), 2579–2605.

[MH18] McInnes L., Healy J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints* (Feb. 2018).

[OK88] Oja E., Kohonen T.: The subspace learning algorithm as a formalism for pattern recognition and neural networks. In *IEEE 1988 International Conference on Neural Networks* (July 1988), vol. 1, pp. 277–284.

[RFS*13] Romero V., Fornés A., Serrano N., Sánchez J. A., Toselli A. H., Frinken V., Vidal E., Lladós J.: The ESPOSALLES database: An ancient marriage license corpus for off-line handwriting recognition. *Pattern Recognition 46*, 6 (2013), 1658–1669.

[SF16] Sudholt S., Fink G. A.: Phocnet: A deep convolutional neural network for word spotting in handwritten documents. In *ICFHR* (2016), IEEE Computer Society, pp. 277–282.

[Sha18] Shapshak P.: Artificial intelligence and brain. *Bioinformation 14*, 1 (2018), 38.

[VGCVH*15] Van Gassen S., Callebaut B., Van Helden M. J., Lambrecht B. N., Demeester P., Dhaene T., Saeys Y.: Flowsom: Using self-organizing maps for visualization and interpretation of cytometry data. *Cytometry Part A 87*, 7 (2015), 636–645.

[VH17] Vats E., Hast A.: On-the-fly historical handwritten text annotation. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on* (2017), vol. 8, IEEE, pp. 10–14.

[VHS17] Vats E., Hast A., Singh P.: Automatic document image binarization using bayesian optimization. In *Proceedings of the 4th International Workshop on Historical Document Imaging and Processing* (2017), ACM, pp. 89–94.

[WLK*67] Watanabe W., Lambert P. F., Kulikowski C. A., Buxto J. L., Walker R.: Evaluation and selection of variables in pattern recognition. In *Computer and Information Sciences* (1967), Tou J., (Ed.), vol. 2, New York: Academic Press, pp. 91–122.

[WP73] Watanabe S., Pakvasa N.: Subspace method in pattern recognition. In *1st Int. J. Conference on Pattern Recognition, Washington DC* (1973), pp. 25–32.

[XRV17] Xiao H., Rasul K., Vollgraf R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR abs/1708.07747* (2017).