Preliminary Results on Ancient Cham Glyph Recognition from Cham Inscription images

Minh-Thang Nguyen[1], Anne-Valérie Shweyer[2], Thi-Lan Le[1], Thanh-Hai Tran[1] and Hai Vu[1]
[1]Computer Vision Department, MICA International Research Institute,
Hanoi University of Science and Technology, Vietnam
[2]French National Center for Scientific Research, France
Email: [1]thang.nm153518@sis.hust.edu.vn, [2]Anne-valerie.SCHWEYER@cnrs.fr,

***Abstract*—This paper presents an original work on ancient Cham glyph which is a language of Cham people in the Southeast Asia from the 6th to 15th century. Unfortunately it is in danger of being destroyed by times as well as of being ignored when the specialists of ancient Cham disappear. Motivated by this fact, we contribute to build a corpus of ancient Cham glyph that contains of 1607 images of Cham inscriptions. These images have been carefully pre-processed then annotated into 37 classes by a specialist/historian in ancient Cham. This digitized version of Cham glyph could be stored anywhere as long as wanted and available to wide audience for studying and exploration. As the corpus is the first introduced world wide, no work on automatic Cham recognition has been considered. We then investigate computer vision and machine learning techniques to show how effective a machine learning technique could be for this case study on a still limited amount of data. The best recognition F1-score is 86.3% with features extracted from GoogleNet and K-NN classifier, showing very promising performance.**

I. Introduction

In recent years, digitizing ancient texts or historical/cultural documents/artifacts has been becoming very active. This process generates digital versions of "analog forms" that could be conserved anywhere, as long as wanted and avoid from being destroyed by times. In addition, digitization makes a big benefit to a wide audience for studying or exploring invaluable documents, searching and enriching them by annotation, classification or recognition.

In this paper, we focus on presenting our original works on digitizing ancient Cham glyphs and investigating how computer vision and machine learning techniques could be useful for automatic recognition of Cham glyphs. Cham is the language of Cham people in the Southeast Asia, and formerly the language of the kingdom of Champa in central Vietnam. Ancient Cham was created from the 6th century then disappeared during the 15th century. Nowadays, ancient Cham has been found on inscriptions on stones at some museums in Vietnam which are being destroyed / abrasive by the time.

Besides, today only a handful of specialists in the world are able to read the inscriptions and when these specialists disappear, no one will be able to read these historical documents. A whole part of Vietnam's history will then be in danger of disappearing. Since Vietnamese writing was romanized in 1920, all ancient writings are destined to disappear due to the lack of practice of different alphabets. As a consequence, conservation, annotation and automatic recognition of ancient Cham documents / artifacts is essential. More than working on a dictionary of Cham words in the inscriptions, it is crucial to be able to create a system of automatic reading of the inscriptions. Every year, archaeological discoveries bright to light new inscriptions, one needs a tool available for the future generation. While a dictionary will be becoming obsolete as discoveries are made, the possibility of automatic reading will adapt to the evolution of the corpus, and provide research tools for the future generations of researchers.

This paper makes two main contributions. Firstly, we build a corpus of ancient Champ glyphs by taking images of Cham inscriptions. These images are then annotated by a specialist/historian of ancient Cham. To the best of our knowledge, it is the first corpus of ancient Cham glyphs worldwide and this could be very useful for historical and linguistic exploration. Second, we deploy machine learning techniques to automatically classify these glyphs into 37 classes with promising performance. Although a number of methods for ancient script recognition from images using computer vision and machines have been proposed in literature [1], [2], [3], nobody has been working on the Cham script. On one hand, there are no dataset of written Cham language. On the other hand, Cham has different characteristic compared to other logographic systems. Many challenges should be faced such as various writing style of same character, overlapping characters, touching characters and various composite characters.

Ancient script recognition could be approached by hand crafted features based methods or deep learning methods [1]. Hand crafted features based methods extracted good features such as shape context, histogram of oriented gradient, local binary patterns, kernel descriptors, that have been manually designed a priori. These features are good for some specific objects but difficult to be generalized to deal with variations of objects and scenes. Deep learning takes benefits of available data to learn automatically features. However deep learning requires a big amount of data. In our case study, even we collect the first dataset of Cham inscription images, as the people working on this topic is very limited, the dataset is still limited at the beginning. Our main question is how knowledge from other data types could be transferred to better understanding of the new data (i.e. Cham glyphs). This work will investigate some existing techniques belonging to both above approaches and produce preliminary results of

978-1-7281-1829-1/19/$31.00 ©2019 IEEE

recognition. The results shows the promising performance of deep learning based method for Cham glyph recognition.

The remaining of this paper is organized as follows. In section II, we present existing works related to ancient scripts recognition. We then describe our new corpus of ancient Cham glyph in section III. The recognition framework is proposed in section IV and their performances are evaluated in section V. Finally we conclude and give some ideas for future works.

II. Related work

In this section, we will present first some existing works for Cham language digitization. We then present techniques for ancient script recognition from images that could be considered for Cham glyph recognition.

A. Cham glyph digitization

Compared to other ancient languages, Cham has been considered less. In 2009, the French School of Asian Studies (École française d'Extrême-Orient, EFEO) launched the project *Corpus of the Inscriptions of Campā* (CIC), aiming to renew the tradition of scholarship on these inscriptions that had thrived at the institution in the early 20th century. The program on registration of Cham from University of New York[1] proposed to offer methodologies for a "digital epigraphy", but it only worked 2 years and it provided only classical works that disconnect texts from their context. Therefore, there are very limited digital resources about ancient Cham until now.

B. Techniques for ancient script recognition from images

As part of the AMADI project funded by STIC ASIE program, researchers from different countries have introduced benchmark datasets and corresponding tasks on ancient languages (Balinese, Sundanese and Khmer) [1], [2], [3]. According to these papers, script recognition from images consists of four main steps: *Binarization; Text line segmentation; Isolated character/glyph recognition*; *Word Recognition and Transliteration*. In the scope of the paper, we focus on the *Isolated character/glyph recognition* task. In this work we will also assume that glyphs are separated in the pre-processing step by an available tool, i.e, we have access to images of cropped glyphs. These works are based on scriptures belong to the same families as the Cham and will be used for the analysis of Cham writings. However, specific investigation is needed to take into account the particularity of the Cham. *Kesiman et al* have conducted a comparison in 29 different schemes for handcrafted feature extraction methods and one self-defined convolutional neural network. We also analyzed the performance of 3 handcrafted feature extraction methods and one automatic feature extraction with 1024-dimensional feature vectors extracted from fully last fully-connected that are also known as neural codes. In our experiment, we look for the optimal feature method in a small dataset. This research will lead to the creation of robust methods for analyzing Cham documents and measure difference between *Deep Learning* and *Traditional Machine Learning* technique in a limited dataset.

[1]http://isaw.nyu.edu/publications/inscriptions/campa/

III. Characteristics of Cham Script/Inscription and Building of the corpus

A. The history and evolution of Cham alphabet

Cham writing was used in the inscriptions of the kingdoms that flourished in Central Vietnam between the 6th and 15th centuries. Inscriptions are texts engraved on stone, brick or even cult objects made of precious metals such as gold, silver or bronze. These texts are mostly praises addressed to the gods by kings, and their family members; they listed the divine virtues and, by reflection, the divine royal qualities, and they enumerated the gifts made to the deities. The script of Cham inscriptions is a writing system derived from the Indian Brahmi. The Brahmi writing system is the ancestor of Cham writing, as are many writings from Southeast Asia. The Cham used the Brahmi alphabet because they locally adapted India's political-religious system and used Sanskrit, a language from India, to address the Hindu gods, who were honoured in Champa from the 6th to the 15th century. In addition, Cham script was used to transcribe two different languages, Sanskrit and Cham. The Sanskrit language is mainly found between the 6th and 10th centuries, while the Cham language is increasingly used between the 11th and 15th centuries. Then with the disappearance of the Cham kingdoms, the use of the ancient Cham was lost; the language was very steeped in culture and vocabulary from India. The disappearance of Hindu political and social structures has led to the disappearance of the use of engraved inscriptions. When one finds the use of Cham on 16th century manuscripts, one speaks of a mid-Cham, because the writing has changed and the vocabulary shows borrowings from the Vietnamese language or Islamic religious customs with the conversion to Islam of part of the Cham population. All the ancient Hindu culture has gradually disappeared. The modern Cham, spoken and written today by members of the Cham ethnic group living in Binh Thuan and Ninh Thuan provinces, is also different, with a modified writing that is visibly adapted to an evolution in pronunciation and a vocabulary imbued with Vietnamese. For the Chams in diaspora in neighbouring Cambodia, the writing itself is different and the vocabulary is impregnated with Khmer. One then speaks of "Eastern Cham" for Cham script and language in Vietnam and "Western Cham" for Cham script and language in Cambodia.

The base material used in this study was provided by twelve inscriptions dated from the 8th to the 13th century, from different regions of ancient Champa, from the South (Phan rang & Nha Trang) to Central Vietnam (Quang Nam and Thua Thien-Hue provinces). Through this chronological range, it is possible to recognize certain changes in writing over time. However, a larger corpus of images will still be needed to determine whether it is possible to isolate writing styles by region or even by craftsmen. At this stage, we have already seen that the evolution of character writing is discontinuous and there are no key periods for a radical change in writing,

but rather a gradual and irregular evolution. For example, as shown in Fig. 1 an "archaic" form of the 8th century letter *ka* will disappear from the 9th century to evolve towards a simpler form. While the archaic form of the letter *ra* will remain until the second half of the 10th century before being simplified Fig. 2.

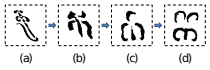

Fig. 1: A letter form of *ka* from a) the early 8th century. b) the late 8th century. c) the middle of 10th century. d) the 13th century.

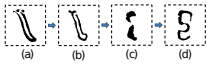

Fig. 2: A letter form of *ra* from a) the 8th century. b) the early 10th century. c) the middle of 10th century. d) the 13th century.

Overall, all characters show a variation over time that probably follows changes in writing styles, such as *śa* in Fig. 3A, *ma* in Fig. 3B, or *na* in Fig. 3C.

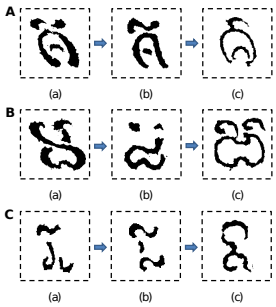

Fig. 3: Various writing style of same character.

B. Characteristic of Cham script

The Cham handwriting of inscriptions followed the Sanskrit notation system, according to the Sanskrit alphabet and the principles of syllabic writing. Thus, each graphical representation - or glyph - is based on one -or more- consonant around which is added a simple additional sign to note a vowel. This system reflects the fact that a consonant cannot be pronounced without a supporting vowel. For instance, the first letter of the Cham alphabet is *ka*, without any additional signs, it included the basic vowel –a; additional signs are used for other vowel notation. Fig. 4 provides an illustration of it.

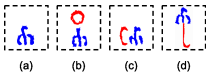

Fig. 4: The first letter of the Cham alphabet ka (a), ki (b), ke (c), and ku (d) (blue: consonant, red: vowel).

The ancient Cham alphabet includes 31 consonants and 11 vowels or diphthongs, which are a, ā, i, ī, e, u, ū, o, au, ai and . Writing system is based on phonology, so the graphemes are classified according to their articulation modes.

C. Cham glyph dataset

After several tests, we decide to not chose to develop our ground truth from the inscriptions themselves on stone or bricks or even photos of these inscriptions, but from photos of rubbings. We chose to work on the images of the rubbings, which are negative inks of the inscriptions; the contrast was then more easily usable for binarization.

To create the word annotated ground truth dataset of the inscriptions, we have been instructed by a research historian in the field of Cham civilization and language. In this research, We have automatically conducted a local binarization technique for inscription images. We have worked together to segment and to manually annotate the isolated character in inscriptions with *ImageJ*, a public domain Java image processing program [4]. The dataset contains 37 classes with 1607 glyphs.

The establishment of the data set encountered several difficulties at the beginning, on the one hand because of the large amount of noise on the images, on the other hand in the precise separation of the lines, and finally in the clear recognition of each syllabic group. The noise cleaning was done manually, as we were unable to apply an automatic cleaning system until the complexity of character notation was resolved. Clearly separating the lines meant precisely recognizing each glyph in its complexity, whether it was a simple consonant with the basic vowel -a (like *ma*) or a consonant group with a diphthong (like *rmmai*). The engravers enjoyed a certain freedom in their work, in particular in the notation of vowels, and their writings occupied a large space between the lines, both above and below. Thus, the vowel -e can be noted the left hand side or the top of the consonant, while the vowel – can be noted the right hand side or the top of the consonant as well, but in a different way.

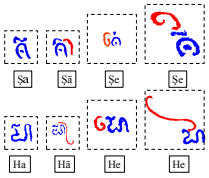

Fig. 5: The engraving style might be affected by variation of writing styles. (blue: consonant, red: vowel)

The same freedom applies to the engraving of consonants, which can interlock for decorative reasons and make it difficult to separate letters for automatic reading. The establishment of a large database to enrich the machine learning system will take time due to the difficulties encountered during the separation of lines and glyphs.

IV. Proposed method for recognizing Cham glyphs from inscription images

A. Overall framework

Figure 6 the overall framework for Cham glyph recognition that consists of two main steps: feature extraction and classification. Two kinds of features that are hand designed features including Histogram of Gradient (HOG) and Neighborhood Pixels Weights (NPW) and learnt features from GoogleNet are used to represent the glyph images. We evaluate these features with two common classification methods that are k-Nearest-Neighbours (KNN) and Support Vector Machine (SVM) . It is worth to note that to build a fully automatic system for Cham inscription understanding, others steps including image binarization, text line segmentation and glyph separation are needed. However, this paper focuses on investigating the performance of the glyph recognition with the assumption is that the others steps have been done.

B. Feature extraction

To represent the Cham glyph images, different features proposed in the state of the art methods can be used. Among these features, HoG and NPW are chosen thanks to their robustness for Southeast Asia palm leaf manuscripts recognition [1]. Besides these hand designed features, CNN feature that has obtained very impressive results on different object recognition tasks is also investigated.

Concerning hand designed features, NPW features are extracted from edge maps by applying the Kirsch directional edge operator [5]. Kirsch defines a non-linear edge enhancement algorithm. Given an image of glyph, 4 images corresponding to 4 directions (horizontal, vertical, left diagonal, right diagonal) are generated (see Fig. 7). Some authors who used/studied this feature for handwritten recognition are: Wen [6], Kim [7] , Knerr et al [8] and Chao [9]. Then NPW feature that measures the distribution of black and white pixels representing various strokes in a glyph image by computing the weights on all the four corners on a pixel due to its neighboring pixels.

HoG feature introduced in [10] has shown its performance in different object recognition tasks. The HOG descriptor is originally defined as the distribution of the local intensity gradients from an image, which are computed from small connected regions (cells). In our paper, the HOG descriptor is calculated over rectangular blocks (R-HOG) with non-overlapping blocks. To ignore negative gradient directions, the range of gradient orientations is defined between 0° and 180° ([10], [11]). The gradient magnitude M and the gradient orientation θ of a pixel at (x,y) position are calculated by

$$M(x,y) = \sqrt{G_x^2 + G_y^2} \tag{1}$$

where G_x and G_y are the horizontal and vertical components of the gradients, respectively.

$$\theta(x,y) = \tan^{-1}\frac{G_x}{G_y} \tag{2}$$

After this, histograms are computed from the occurrences of oriented gradients across large structures (blocks) of the image as shown in Fig. 8. The gradient orientations are stored into 9 orientation bins β. The combination of the histograms from each block represents the feature descriptor. The feature vector size of the HOG descriptor depends on the selected numbers of blocks and bins. It has been shown that the performance of the HOG descriptor depends mostly on the number of blocks ([12]).

Fig. 7: First sub-image (left side) is original image, four images generated by using four Kirsch operators.

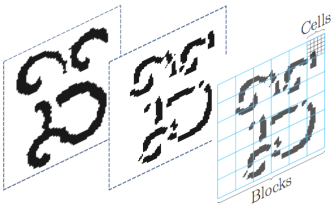

Fig. 8: Example of the rectangular HOG descriptor.

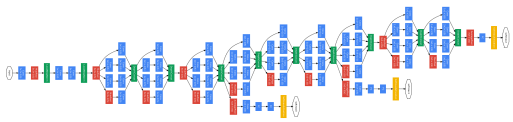

Fig. 9: An illustration of Googlenet's architecture. (adapted from [13])

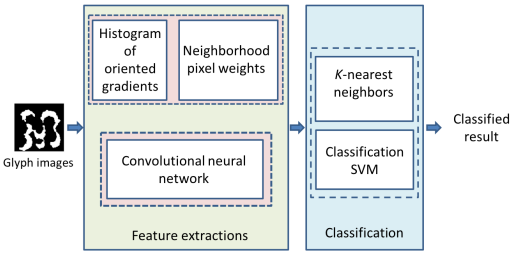

Fig. 6: The proposed framework for Cham glyph image recognition

Concerning deep learning-based features, recently, a numereous CNN architectures have been proposed for object recognition such as Lenet, Alexnet, GoogleNet, Resnet. In this work, we used GoogleNet that has won ILSVRC 2014 as a feature extractor [13]. Fig.9 shows a schematic view of GoogleNet. It is a very deep neural network model with 22 layers when counting only layers with parameters (or 27

layers if we also count pooling). The overall number of layers (independent building blocks) used for the construction of the network is about 100. GoogleNet incorporates *Inception* module with the intention of increasing network depth with computational efficiency. For each glyph image, a vector of 1024 dimensions is extracted from GoogleNet before fully connected layer.

C. Classification

Two common classification methods that are kNN and SVM [14] [15] are chosen.

The *k*-Nearest-Neighbours (*k*NN) is a non-parametric classification method, which is simple but effective in many cases [16]. For a data record *t* to be classified, its *k* nearest neighbours are retrieved, and this forms a neighbourhood of *t*. Majority voting among the data records in the neighbourhood is usually used to decide the classification for *t* with or without consideration of distance-based weighting. However, to apply *k*NN we need to choose an appropriate value for *k*. In our work, k is empirically determined.

The basic idea of SVM is to find an optimal hyper-plane for linearly separable patterns in a high dimensional space where features are mapped onto. There is more than one hyper-plane satisfying this criterion. The task is to detect the one that maximizes the margin around the separating hyper-plane. This finding is based on the support vectors which are the data points that lie closest to the decision surface and have direct bearing on the optimum location of the decision surface. SVMs are extended to classify patterns that are not linearly separable by transformations of original data into new space using kernel function into a higher dimensional space where classes become linearly separable.

V. Experimental results

The experiments have been performed on the Cham glyph dataset that is described in the previous section. The dataset contains 1607 images of 37 glyphs. Figure 10 shows the distribution of the number of samples for each class. As shown in Figure, the working dataset is unbalanced because of its appearance frequency in the Cham inscriptions. For each class, 70% of samples are used for training and the remaining are used for testing.

To evaluate the performance of the proposed framework for ancient Cham glyph recognition, we use F1 score that is defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

$$\text{F1 score} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{5}$$

where TP: True Positive, TN: True Negative; FN: False Negative.

With 3 features and 2 classification methods, 6 different schemes are investigated and evaluated. The obtained results are shown in Tab. I and Fig. 11. Several observations can be given as follows. First, among the three features used for glyph image representation, the learning-based feature with GoogleNet architecture outperforms the hand designed features for Cham glyph recognition. We can see using only NPW could not distinguish the Cham glyph. The F1 scores when using this feature with both classification methods are relatively low and can not acceptable. Secondly, despite the fact that kNN is simple, the performance of this classification is better than that of SVM for deep learning based feature. In case of using HoG and NPW, this classification obtains competitive results in comparison with SVM. Finally, even the Cham glyph recognition is challenging, the obtained results with 86.3% of F1 score for the best case are promising. This means that deploying automatic Cham glyph image recognition is feasible. To better understand the behavior of the proposed framework, we analyze some incorrect recognition cases in Fig. 12. In most cases, the noise is the most important factor that lead to wrong recognition. Due to the noise, some parts of glyph are lost or added. In some cases, due to the large inter-class similarity, the glyph recognition is very challenging even for human being.

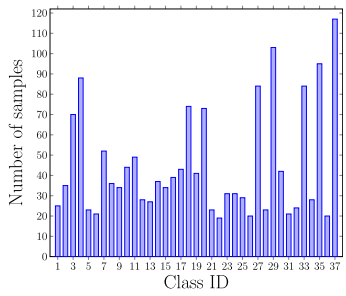

Fig. 10: Number of samples in each class of the dataset.

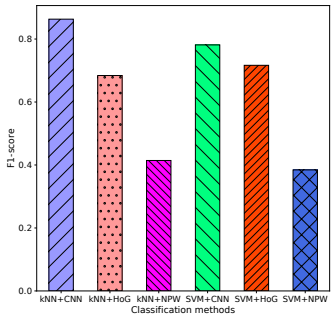

Fig. 11: F1 scores obtained with different features and classification methods for Cham glyph images without noises.

VI. Conclusions and Future works

In this paper, we have presented our work dedicated to Cham glyph recognition in Cham inscription. To the best of our knowledge, this is the first work for Cham glyph recognition with two main contributions: building an image dataset

	Features		
Classification	CNN	HOG	NPW
K-NN	**0.863**	0.684	0.414
SVM	**0.781**	0.717	0.384

TABLE I: F1 score obtained on six investigated schemes for Cham glyph recognition in the test set without noise.

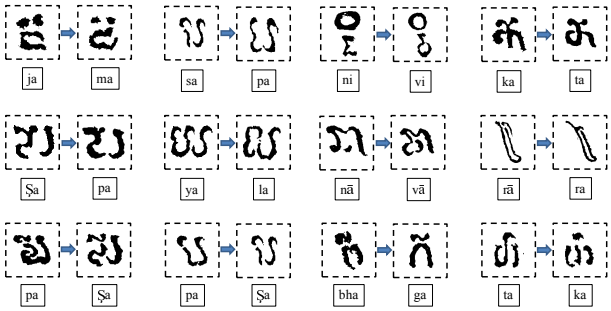

Fig. 12: Examples of incorrect classified glyphs.

of Champ glyph and investigating the performance of state of the art features and classification methods for Cham glyph recognition. As results, a dataset of 37 Cham glyphs with 1607 images has been created. This dataset is progressing steadily and could reach up several hundred thousand images in the near future. The experimental results confirm the feasibility of automatic Cham glyph recognition. In the future, we will extend our work to take into account others steps such as text line separation.

VII. Acknowledgement

The authors would like to thank for the Centre National de la Recherche Scientifique (CNRS) that supported this project. This is the programme "Aide à la Mobilité Internationale" and the DERCI (Direction Europe de la Recherche et Coopération Internationale).

References

[1] M. W. A. Kesiman, D. Valy, J.-C. Burie, E. Paulus, M. Suryani, S. Hadi, M. Verleysen, S. Chhun, and J.-M. Ogier, "Benchmarking of document image analysis tasks for palm leaf manuscripts from southeast asia," *Journal of Imaging*, vol. 4, no. 2, p. 43, 2018.

[2] A. Kesiman, M. Windu, J.-C. Burie, J.-M. Ogier, and P. Grangé, "Knowledge representation and phonological rules for the automatic transliteration of balinese script on palm leaf manuscript," *Computación y Sistemas*, vol. 21, no. 4, pp. 739–747, 2017.

[3] M. W. Kesiman, D. Valy, J.-C. Burie, E. Paulus, I. M. G. Sunarya, S. Hadi, K. H. Sok, and J.-M. Ogier, "Southeast asian palm leaf manuscript images: a review of handwritten text line segmentation methods and new challenges," *Journal of Electronic Imaging*, vol. 26, no. 1, p. 011011, 2016.

[4] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid *et al.*, "Fiji: an open-source platform for biological-image analysis," *Nature methods*, vol. 9, no. 7, p. 676, 2012.

[5] S. Kumar, "Neighborhood pixels weights-a new feature extractor," *International Journal of Computer Theory and Engineering*, vol. 2, no. 1, p. 69, 2010.

[6] Y. Wen, Y. Lu, and P. Shi, "Handwritten bangla numeral recognition system and its application to postal automation," *Pattern recognition*, vol. 40, no. 1, pp. 99–107, 2007.

[7] K. M. Kim, J. J. Park, Y. G. Song, I. C. Kim, and C. Y. Suen, "Recognition of handwritten numerals using a combined classifier with hybrid features," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, 2004, pp. 992–1000.

[8] S. Knerr, L. Personnaz, and G. Dreyfus, "Handwritten digit recognition by neural networks with single-layer training," *IEEE Transactions on neural networks*, vol. 3, no. 6, pp. 962–968, 1992.

[9] S.-B. Cho, "Neural-network classifiers for recognizing totally unconstrained handwritten numerals," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 43–53, 1997.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[11] J. Arróspide, L. Salgado, and M. Camplani, "Image-based on-road vehicle detection using cost-effective histograms of oriented gradients," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 1182–1190, 2013.

[12] O. Déniz, G. Bueno, J. Salido, and F. De la Torre, "Face recognition using histograms of oriented gradients," *Pattern Recognition Letters*, vol. 32, no. 12, pp. 1598–1603, 2011.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[14] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: Primal estimated sub-gradient solver for svm," *Mathematical programming*, vol. 127, no. 1, pp. 3–30, 2011.

[15] M. Takác, A. S. Bijral, P. Richtárik, and N. Srebro, "Mini-batch primal and dual methods for svms." in *ICML (3)*, 2013, pp. 1022–1030.

[16] D. Hand, H. Mannila, and P. Smyth., *Principles of Data Mining*. The MIT Press, 2001.