VML-HP: Hebrew paleography dataset

Ahmad Droby[1], Berat Kurar Barakat[1], Daria Vasyutinsky Shapira[1], Irina Rabaev[2], and Jihad El-Sana[1]

[1] Ben-Gurion University of the Negev, Beer-Sheva, Israel
[2] {drobya,berat,dariavas}@post.bgu.ac.il
el-sana@cs.bgu.ac.il
[3] Shamoon College of Engineering, Beer-Sheva, Israel
irinar@ac.sce.ac.il

Abstract. This paper presents a public dataset, VML-HP, for Hebrew paleography analysis. The VML-HP dataset consists of 537 document page images with labels of 15 script sub-types. Ground truth is manually created by a Hebrew paleographer at a page level. In addition, we propose a patch generation tool for extracting patches that contain an approximately equal number of text lines no matter the variety of font sizes. The VML-HP dataset contains a train set and two test sets. The first is a typical test set, and the second is a blind test set for evaluating algorithms in a more challenging setting. We have evaluated several deep learning classifiers on both of the test sets. The results show that convolutional networks can classify Hebrew script sub-types on a typical test set with accuracy much higher than the accuracy on the blind test.

Keywords: Paleography · Handwritten style analysis · Hebrew medieval manuscripts · Script type classification· Learning-based classification · Convolutional Neural Network

1 Introduction

Robust and accurate algorithms in document image analysis can be developed and compared by the public availability of labeled datasets. A vital document image analysis task is to provide solutions for the study of ancient and medieval handwriting.

Paleography (from Greek "palaios" - "old" and "graphein" - "to write") is a study of handwriting. Throughout history, different script types were used for different types of manuscripts; these script types appeared, developed, and disappeared as time went by. The classification of script types started in the middle ages, and the contemporary paleography research of Latin, Greek, and Hebrew scripts emerged in the mid-20th century. An experienced librarian who works with medieval or ancient manuscripts knows to recognize and read the scripts of a given collection. A researcher specially trained to recognize and compare all medieval and ancient script types and sub-types is called a paleographer. The paleographic analysis is used to determine the place and date of writing manuscripts that have no date, fragmentary and damaged manuscripts, etc. In some cases, it is possible, by comparison, to identify the scribe, to check the manuscripts' authenticity, or derive other essential data.

Contemporary Hebrew paleography emerged in the mid-1950s, concurrently with modern Latin paleography. The theoretical basis of Hebrew paleography is formulated in the works of Malachi Beit-Arié, Norman Golb, Benjamin Richler, Colette Sirat [2,3,23,27,20,19]. Hebrew manuscripts have a stereotyped nature of handwriting, which is a product of the cultural and pedagogical convention. A scribe was required to emulate the writing of his master until the forms of their writing become indistinguishable. This stereotypical script of a specific region is called a script type; script sub-types reflect the time of writing the manuscript or its type, whereas the separation of the hands within a single script sub-type is referred to as a unique handwriting style.

The digital era has enabled Hebrew manuscript images to be accessible publicly. This paper exploits these manuscript images to introduce the first publicly available Hebrew paleography dataset called VML-HP (Visual Media Lab - Hebrew Paleography). We believe that Hebrew paleography dataset is an important resource for developing a large-scale paleographic analysis of Hebrew manuscripts as well as for evaluating and benchmarking the analysis of algorithms for script classification. A trained paleographer can only describe a limited number of manuscripts, the number of such paleographers is very small, and there are still manuscript collections without even a good basic catalogue. We believe that a proper algorithm will become an essential tool in manuscripts' research.

Contemporary Hebrew paleography identifies six main-types of scripts: Byzantine, Oriental, Yemenite, Ashkenazi, Italian, Sephardic. Each main script type may contain up to three sub-types of scripts: square, semi-square, cursive (Fig. 1). In total, there are 15 script sub-types, which are included in the VML-HP dataset. Fig. 2 shows example document patches for each Hebrew script subtype from the dataset. The VML-HP dataset contains a train set and two test sets. The first test set is called the typical test set and is composed of unseen pages from the manuscripts used in the train set. The second test set is called the blind test set and is composed of unseen pages from the manuscripts not used in the train set. Currently, the VML-HP dataset can be downloaded from `http://www.cs.bgu.ac.il/~berat/`.

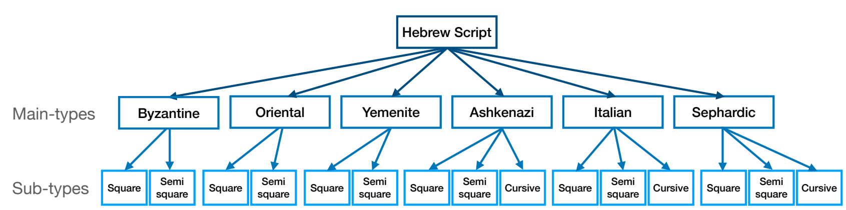

Fig. 1. Medieval Hebrew script has six main-types, and each main-type has up to three sub-types. In total, there are 15 sub-types of Hebrew script.

Fig. 2. Example document image patches of the 15 Hebrew script types.

In this paper, we investigate two problems: (1) the classification of the 15 script types available in the VML-HP dataset, and (2) the distinction between square and cursive scripts, regardless of the script main-type. Distinguishing between square and cursive scripts can help to identify important sections of a manuscript, which are often written in a different script sub-type. Such sections include colophon (an inscription at the end, less often at other places, of the manuscript that tells when, where, and by whom it was written), the owner's notes and more. Since colophon and other text notes are the primary sources of information about the manuscript, knowing exactly where they are located on the page would be a great help for a researcher.

We provide baseline results for the two classification problems using several convolutional neural networks. The networks are trained to classify patches that are extracted from the pages in the VML-HP dataset. Each network is evaluated on the blind and typical test sets at patch level and page level, where a page is classified based on the majority vote of its patches. Unsurprisingly, all of the networks achieved significantly higher accuracy on the typical test set compared to the blind test set, with ResNet50 giving the best performance. In addition, we explore preprocessing the input patches and applying different data augmentation strategies to improve the results.

The rest of the paper is organized as follows: Section 2 is a short survey of the related literature. Section 3 describes the theoretic foundation, collection and properties of the VML-HP dataset. Section 4 explains two ground truth formats provided with the dataset. In Section 5 we evaluate several deep learning classifiers on VML-HP.

2 Related Work

During the last decade, various computer vision techniques were employed for paleography analysis. Earlier methods used manually defined features, mostly based on textural, grapheme-based descriptors, and their combination [14,13,12].

Over the recent decade, deep learning methods have achieved new standards in many research frontiers. The early work [10,4] trained CNNs to classify the writing styles and used the penultimate layer activations as features. Such supervised methods require a lot of labeled data. Christlein *et al.*[5] and Hosoe *et al.*[15] showed that deep activations learned in an unsupervised manner can perform better. In a writing style classification dataset, the training classes are different from the test ones. To handle this difference, Keglevic *et al.*[17] propose to use a triplet CNN that measures the similarity of two image patches instead of training a classification network on surrogate classes. The work of Abdalhaleem *et al.*[1] examines the in-writer variations in a manuscript. Their model is based on Siamese convolutional neural networks, where the model is trained to learn the tiny changes in a person's writing style.

Deep learning methods won first place in the competitions on the classification of medieval handwritings in Latin script [8,7] held in 2016 and 2017. The objective of the competition was to classify medieval Latin scripts into 12 classes according to their writing style. Also, another level of classification - the date classification - was added in [7]. The results show that deep learning models can classify Latin script types with acceptable accuracy, more than 80% on homogeneous document collections (TIFF format only) and about 60% on heterogeneous documents (JPEG and TIFF images). Studer *et al.* [24] studied the effect of ImageNet pre-training for different historical document analysis tasks, including style classification of Latin manuscripts. They investigated a number of well-known architectures: VGG19 [22], Inception V3 [25], ResNet152 [11] and DenseNET12 [16]. The models were trained from scratch and obtained 39% to 46% accuracy rate on the script classification task, while the pre-trained models obtained a $49\% - 55\%$ accuracy rate.

Early research on the paleographic classification of Hebrew documents is described in [26], where authors apply computerized tools on the documents from the Rabbanite Cairo Genizah collection. They construct a dictionary based on k-means clustering and represent each document using a set of descriptors based on the constructed prototypes. Dot product is applied between the descriptors of two documents to measure their similarity. The results depend crucially on the methods used to construct the dictionary. Dhali *et al.* [9] apply textural and grapheme-based features together with support vector regression to estimate a date of the ancient manuscripts from Dead Sea Scrolls collection.

This paper utilizes deep learning models to classify medieval Hebrew manuscripts according to their script type. Our dataset mainly includes manuscripts from SfarData collection (see Section 3). In some rare cases, we added manuscripts from other collections to balance the number of representative documents in each class.

3 Construction of VML-HP

VML-HP dataset is built upon a theoretic foundation of contemporary Hebrew paleography. It contains 537 accurately labeled, high-resolution manuscript page images that are hierarchically organized into six main-types and 15 sub-types of Hebrew script (Fig. 1). A Hebrew paleographer collected and labeled these manuscript page images manually.

3.1 SfarData

SfarData[4], an ongoing database project based on the Hebrew Paleography project, in cooperation with the Israeli Academy of Sciences and Humanities and the National French Center for Scientific Research (CNRS), was initiated by Malachi Beit-Arié in the 1970s. Paleograpic and codicological creteria of our project are derived from this site. Sfardata aims to locate, classify, and identify all existing dated Hebrew manuscripts written before 1540. Today it includes almost 5000 manuscripts, which makes about 95% of the known dated medieval Hebrew manuscripts. This database is currently hosted at the site of the National Library of Israel. It includes the codicological and paleographical features of the manuscripts obtained *in situ*, i.e., in the libraries in which they are kept. The project intends to study and classify these features to expose the historical typology of Hebrew manuscripts.

The most important collection of digitized and microfilmed Hebrew manuscripts belongs to the Institute for Microfilmed Hebrew manuscripts at the National Library of Israel. The Institute has been collecting microfilms (now digital photos) of Jewish manuscripts for decades, and its goal is to obtain digital copies of all Hebrew manuscripts worldwide. Today, the Institute hosts more than 70,000 microfilms and thousands of digital images, which makes more than 90% of the known Hebrew manuscripts in the world.

3.2 Collecting manuscript pages

Initially, the primary criterion for selecting manuscripts for our project was the fact that they were described in the SfarData. In cases when this turned out to be impossible, we chose manuscripts based on the SfarData criteria. Two subtypes proved to be particularly problematic: the Oriental square and Ashkenazi cursive. The Oriental square is the oldest Hebrew script sub-type. Most manuscripts written in this script type are not complete (collection of fragments). The better-preserved ones are kept in the National Library of Russia (Firkowicz manuscripts' collections), whose collections have not yet been entirely digitized. The Ashkenazi cursive, on the other hand, is a very common script with lots of manuscripts. However, it developed and began to be actively used only shortly before 1540, and thus there are not enough examples in the SfarData.

[4] http://sfardata.nli.org.il/

Pages in the VML-HP dataset were extracted from high-quality digitized manuscripts. Among the manuscripts described in SfarData, we gave first preference to those kept in the National Library of Israel. When this was impossible, we used manuscripts from other libraries, first and foremost the British Library and the Bibliothèque Nationale de France, which have vast collections of digitized manuscripts available for download. Whenever good quality digital photos of manuscripts were not available or were of insufficient quantity for some Hebrew script sub-types, we turned to microfilms from the collection of the Institute for Microfilmed Hebrew manuscripts at the National Library of Israel.

The initial dataset that is described in this paper is relatively small. The reason is that all the manuscripts were manually picked up by our team's paleographer. Each chosen manuscript, except that it met the requirements described in the previous paragraph, had to be written in a typical (and not deviated) script sub-type that it stood for, had to have one script per page (and not multiple scripts on one page), etc. This meant weeks and even months of work. Also, the number of manuscripts of the required quality and available for download turned out to be very limited.

3.3 Properties of VML-HP

VML-HP aims to provide complete coverage of the Hebrew paleography study. It contains accurately labeled page images for each of the 15 sub-type scripts. The VML-HP dataset contains a train set and two test sets. The first test set is called the typical test set and is composed of unseen pages from manuscripts used in the train set. The second test set is called the blind test set and is composed of unseen pages from manuscripts that are not used in the train set. The blind test set is more challenging as it comes from another distribution than the train set's distribution; however, it is necessary for evaluating and benchmarking algorithms in a real-world scenario. Table 1 shows the distributions of the number of pages per main-type and sub-type scripts in the train set and two test sets. VML-HP is constructed with the goal that all the discriminator features of a script type are included in the page images with all possible variable appearances. To our knowledge, this is the first accurately labeled Hebrew paleography dataset available to the document image analysis research community.

4 Ground truth of VML-HP

VML-HP dataset images are accurately labeled by a Hebrew paleographer at page level. However, page level labels are not always fully useful for a computer algorithm because historical document images suffer from several issues, such as physical degradation, ink bleed through, ink degradation, and image noise. In addition, hand painted motifs commonly appear in mediaval manuscripts. Therefore, we provide a clean patch generation tool that generates image patches with approximately five text lines. And we include additional ground truth format which stores the coordinates of bounding polygons around

Main-Type	Sub-Type	Train	Typical Test	Blind Test	Total
Ashkenazi	Square	16	4	10	30
	Semi-Square	16	3	10	29
	Cursive	16	4	10	30
	Total	48	11	30	89
Byzantine	Square	16	4	10	30
	Semi-Square	16	4	10	30
	Total	32	8	20	60
Italian	Square	16	4	10	30
	Semi-Square	16	4	10	30
	Cursive	16	4	10	30
	Total	48	12	30	90
Oriental	Square	64	14	10	98
	Semi-Square	16	4	10	30
	Total	80	18	20	118
Sephardic	Square	16	4	10	30
	Semi-Square	24	6	10	40
	Cursive	16	4	10	30
	Total	56	14	30	100
Yemenite	Square	24	6	10	40
	Semi-Square	24	6	10	40
	Total	48	12	20	80
Total		312	75	150	537

Table 1. The distributions of the number of pages per main-type and sub-type scripts in the train set and two test sets.

the text regions into PAGE-XML [6,18] files. The PAGE-XML files and the clean patch generation tool are available together with the dataset at `http://www.cs.bgu.ac.il/~berat/`.

4.1 Clean patch generation algorithm

Often occurring non-text elements cause a naive patch generation algorithm (that only considers the foreground area) to generate patches with irrelevant or limited features (Fig. 3). Moreover, varying font-sizes across manuscripts may lead to low level cues like the number of text lines or sizes of characters in a patch.

To ensure that the classifier algorithm extracts the desired features, script shape features in our case, we propose a clean patch generation algorithm that can generate patches containing pure text regions and an approximately equal number of text lines. A document image patch needs to include the maximum possible amount of script features while still fitting the memory requirements. According to paleographers, a patch would be sufficient to figure out the script

Fig. 3. Example patches generated by a naive algorithm. Some patches contain irrelevant features, some patches contain only a few characters, and others contain no text at all.

type if it contains approximately five text lines. We first calculate the patch size $s \times s$ that includes approximately five text lines for each page. Then, we randomly generate patches of the size $s \times s$ from this page and resize them to the appropriate size based on the network and memory requirement. Finally, we validate the generated patches according to the additional criteria described below.

Extracting patches with *n* text lines To calculate the patch size s that includes n text lines, we extract k random patches of the size equal to one-tenth of the page height. A patch of this size usually includes several text lines. The desired patch size is given by $s = h/10 \times n/m$, where h is the height of the page, n is the average targeted number of lines, and m is the actual average number of lines in the k extracted patches. The number of lines in a given patch is computed by counting the peaks of the y profile using Savitzky-Golay filter [21]. We used $n = 5$ and $k = 20$.

Patch validation Each extracted patch is validated according to the following conditions:

- The foreground area should be at least 20% of the total patch area and not exceed 70% of the total patch area. This condition eliminates almost empty patches and patches with large spots, stains, or decorations.
- The patch should contain at least 30 connected components. This condition eliminates patches with few foreground elements.
- The variance of the x and y profiles denoted by σ_x and σ_y, respectively, should satisfy the conditions $\sigma_x \leq T_x$, $\sigma_y \geq T_y$. Assuming horizontal text lines, the variance of the x profile should be relatively low. During our experiments we set $T_x = 1500$ and $T_y = 500$
- The following inequality should be satisfied:

$$0.5 \leq \frac{\sum_{i=0}^{\frac{v}{2}} P_x(i)}{\sum_{i=\frac{v}{2}}^{v} P_x(i)} \leq 1.5 \tag{1}$$

  Where v is the number of values in the x profile and $P_x(i)$ is the i-th value. This condition eliminates the patches with text lines that occupy only a fraction of a patch.

4.2 Clean patch generation results

Fig. 4 shows some example output patches from the clean patch generation algorithm, and Fig. 5 illustrates that the generated patches are sampled uniformly over the text regions.

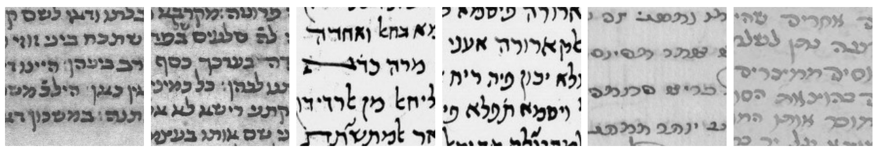

Fig. 4. Example output patches from the clean patch generation algorithm.

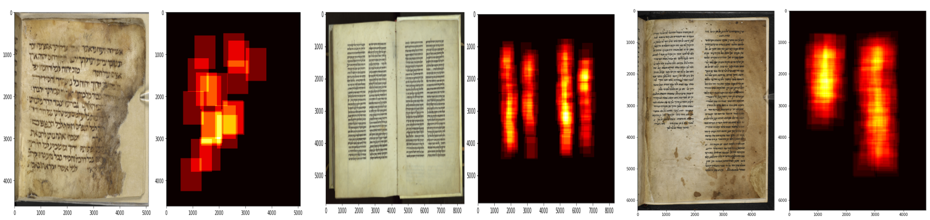

Fig. 5. Heat maps illustrate the location distribution of the generated patches covers the text regions.

5 Evaluation of deep learning classifiers on VML-HP

In this section, we report the results of several deep learning classifiers. These experiments provide baseline results for potential benchmarking and underline that real-world problems are significantly challenging. First, we demonstrate the necessity of having two test sets, the typical test set and the blind test set. Then, we introduce the setting used in all the experiments, followed by evaluating different types of convolutional networks. Finally, we investigate the influence of preprocessing the input patches and the influence of different data augmentation strategies.

5.1 Real world challenge

A dataset is usually split randomly into training, validation, and test sets. In such a scenario, pages belonging to the same manuscript may appear in the training and test sets. While this is a standard scheme, such a split can lead to misleading results. The model can learn to identify features specific to a manuscript, such as a background texture, ink color, or handwriting style. These features, as a whole, can mistakenly be used for classification. To assess this, we created a blind test set containing pages from manuscripts that are not present in the train set.

The necessity of a blind test set is visualized by training a pretrained ResNet50 and embedding the extracted feature vectors onto 2D space using t-SNE. Fig. 6 shows the train, typical test, and blind test set clusters before the training. Fig. 7 shows the train, typical test, and blind test set clusters after the training. The training embeds the train set samples and the typical test set samples onto compact and well-separated clusters relative to the fuzzy clusters of the blind test set samples. This shows the hardness of discovering a pattern among the blind test set samples.

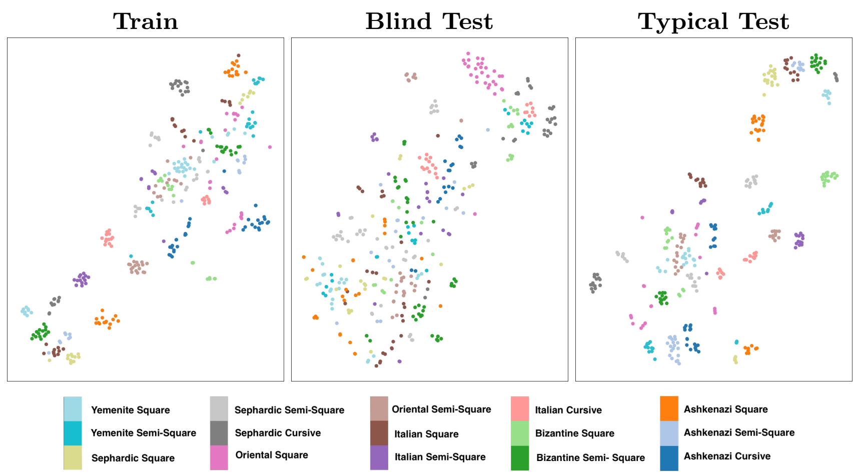

Fig. 6. Initial distribution embedding of the train, typical test and blind test sets before training.

5.2 Experimental setting

We experiment with several convolutional network architectures. In all the experiments, we train the network on the training set and test it on both test sets, typical and blind. First, we generate 150K train patches, 10K typical test patches, and 10K blind test patches of size 350×350 using the clean patch generation algorithm proposed in Section 4.1. Input patches are normalized in terms of their pixel values. The objective training function is cross-entropy loss and is minimized using the Adam optimizer algorithm. We continue training until there is no improvement in validation loss with five epochs' patience and save the model with the least validation loss for testing.

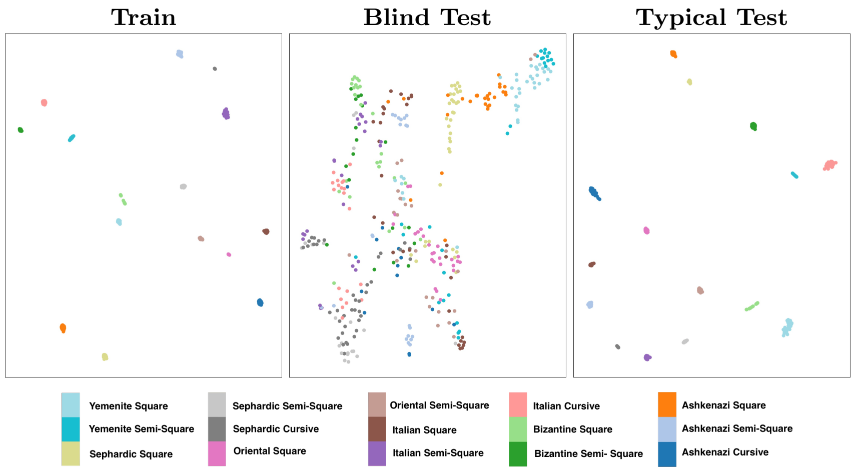

Fig. 7. Final distribution embedding of the train, typical test and blind test sets after training.

Classification results are evaluated by patch and page levels accuracy. For the page level accuracy, the label of a page is computed by taking the majority vote of the predictions of 15 patches from the page.

5.3 Effect of network type

Table 2 shows the accuracy results for classifying 15 script sub-types using different convolutional networks, comparing the typical test set and the blind test set at patch and page levels. The results indicate that the typical test set patches and pages are easier to classify. At nearly all levels and sets, the performance of the ResNet50 classifier is consistently higher but does not surpass 40% accuracy on the blind test set. The difference between the typical test set accuracy and the blind test set accuracy indicates overfitting on irrelevant features. However, the random guess accuracy of 15 classes is 7.6%, indicating that the network extracts script type-dependent features and improves the random classification accuracy. We argue that script type classification is an expressible function, but the network needs more data to learn this function.

Table 3 shows the accuracy results for classifying square and cursive script sub-types using different convolutional networks, comparing the typical test set and the blind test set at patch and page levels. The typical test set accuracy is fully saturated, whereas there is little room for improvement at blind test set accuracy. This result strengthens the above argument because decreasing the number of classes from 15 to two increases the number of samples per class, leading to higher accuracy.

Table 2. Patch and page level accuracies on typical test set and blind test set using different network architectures for classifying 15 script sub-types.

	Patch level		Page level	
	Typical	Blind	Typical	Blind
DenseNet	97.97	32.95	98.63	38.36
AlexNet	91.99	27.03	93.15	28.28
VGG11	99.16	35.55	100	35.63
SqueezeNet	98.03	30.38	98.63	29.45
ResNet18	97.07	30.95	98.63	34.25
ResNet50	99.55	36.15	98.63	39.73
Inception v3	94.94	26.41	95.89	26.71

Table 3. Patch and page level accuracies on typical test set and blind test set using different network architectures for classifying square and cursive script sub-types.

	Patch level		Page level	
	Typical	Blind	Typical	Blind
DenseNet	99.85	87.06	100	83.72
AlexNet	99.48	88.01	100	91.86
VGG11	99.93	86.45	100	88.37
ResNet18	99.65	86.85	100	87.21
ResNet50	99.99	90.58	100	94.19
SqueezeNet	98.03	82.45	100	86.05
Inception v3	99.16	82.06	100	80.23

5.4 Effect of preprocessing

As the irrelevant background features might lead to poor training, we preprocess the patches by applying a bilateral filter and a bandpass filter, reducing the amount of information passed to the network through the background pixels. Hence, improving the overfitting of a convolutional network on spurious background frequencies. Table 4 shows the effect of preprocessing using a ResNet50 network on 15 script sub-types. As validated in the results, preprocessing further boosts the performance to 42.1% and 49.3% accuracy at patch and page levels, respectively.

Table 4. Effect of preprocessing at patch and page level accuracies using a ResNet50 network on blind test set for classifying 15 script sub-types.

Preprocessing	Patch level	Page level
x	36.15%	39.73%
v	42.10%	49.30%

5.5 Effect of augmentation

It is known that augmenting the training data increases the model's ability to overcome overfitting. We experimented using various combinations of augmentation methods, such as random scaling by a factor between 1.0 and 1.2, random rotation by a degree between $-30°$ and $30°$, and random horizontal flipping.

Table 5 shows the effect of augmentation at patch and page levels accuracy results on the blind test set for classifying 15 script sub-types using a ResNet50 network. Random rotation and horizontal flipping boost the accuracy at both the patch and page levels. However, random scaling only improves the patch level accuracy. In addition, we can conclude that applying all augmentation strategies at once is counterproductive. Perhaps it biases the network through the train set distribution, which is very dissimilar to the data distribution that the model is tested on.

Furthermore, we experimented with combining preprocessing and augmentation. However, doing so did not improve the results, and in some cases, it worsened it. We hypothesize that in some cases, combining preprocessing and augmentation results in the loss of text features, which reduced the classification accuracy.

Table 5. Effect of augmentation by patch and page levels accuracy results on blind test set for classifying 15 script sub-types using a ResNet50 network.

Augmentations			Patch level	Page level
Scaling	Rotation	H. Flip		
×	×	×	36.15%	39.73%
✓	×	×	38.76%	32.88%
×	✓	×	37.11%	42.47%
×	×	✓	38.30%	43.84%
✓	✓	✓	32.06%	30.82%

5.6 Comparing deep neural networks against a paleographer expert

To compare the deep learning networks against a paleographer expert, we performed a classification experiment. In this experiment, we ask a paleographer to classify 75 document patches according to the 15 script sub-types. The document patches were randomly chosen from the set of patches used in our experiments, five from each sub-type. The accuracy rate of the paleographer expert is 70%. We can draw two conclusions from this experiment. First, the problem was challenging for the human expert due to the unusual format: paleographers work with manuscripts and pages, not patches. Second, there is a large room for improvement for automatic classification. As we previously pointed, we expect that training the networks on a larger dataset will improve the classification rate on the blind test set. A potential limitation of this experiment is that it was performed only with one paleographer expert. Unfortunately, the number of Hebrew paleographers is extremely small, and we did not want to involve the paleographers who created the SfarData. However, we do not expect a larger experiment to change the results significantly.

6 Conclusion

Automatic paleographic analysis of historical documents is a challenging task, and benchmark datasets lie at the heart of the development, assessment, and comparison of the algorithms. This paper introduces a medieval Hebrew manuscripts dataset, VML-HP dataset, which includes 537 pages labeled with 15 script sub-types. The VML-HP dataset contains a train set, typical and blind test sets. The VML-HP is the first publicly available Hebrew paleographic dataset.

We report baseline results of several established deep learning classification networks. Results show that there is a big room for improvement on the blind test set, whereas the typical test set is an easier mission. In addition, we showed that preprocessing the input patches by applying a bilateral filter and a bandpass filter boosts the model's performance. Furthermore, we explored different data augmentation strategies.

Acknowledgment

This research was partially supported by The Frankel Center for Computer Science at Ben-Gurion University. The participation of Dr. Vasyutinsky Shapira in this project is funded by Israeli Ministry of Science, Technology and Space, Yuval Ne'eman scholarship n. 3-16784.

References

1. Abdalhaleem, A., Barakat, B.K., El-Sana, J.: Case study: Fine writing style classification using siamese neural network. In: 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR). pp. 62–66. IEEE (2018)
2. Beit-Arié, M.: Hebrew codicology. Tentative Typology of Technical Practices Employed in Hebrew Dated Medieval Manuscripts. Jerusalem (1981)
3. Beit-Arié, M., Engel, E.: Specimens of mediaeval Hebrew scripts, in 3 vol. Israel Academy of Sciences and Humanities (1987, 2002, 2017)
4. Christlein, V., Bernecker, D., Maier, A., Angelopoulou, E.: Offline writer identification using convolutional neural network activation features. In: German Conference on Pattern Recognition. pp. 540–552. Springer (2015)
5. Christlein, V., Gropp, M., Fiel, S., Maier, A.: Unsupervised feature learning for writer identification and writer retrieval. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 991–997. IEEE (2017)

6. Clausner, C., Pletschacher, S., Antonacopoulos, A.: Aletheia-an advanced document layout and text ground-truthing system for production environments. In: ICDAR. pp. 48–52. IEEE (2011)
7. Cloppet, F., Eglin, V., Helias-Baron, M., Kieu, C., Vincent, N., Stutzmann, D.: Icdar2017 competition on the classification of medieval handwritings in latin script. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 1371–1376. IEEE (2017)
8. Cloppet, F., Eglin, V., Stutzmann, D., Vincent, N., et al.: Icfhr2016 competition on the classification of medieval handwritings in latin script. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 590–595. IEEE (2016)
9. Dhali, M.A., Jansen, C.N., de Wit, J.W., Schomaker, L.: Feature-extraction methods for historical manuscript dating based on writing style development. Pattern Recognition Letters **131**, 413–420 (2020)
10. Fiel, S., Sablatnig, R.: Writer identification and writer retrieval using the fisher vector on visual vocabularies. In: 12th International Conference on Document Analysis and Recognition. pp. 545–549. IEEE (2013)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. He, S., Samara, P., Burgers, J., Schomaker, L.: Discovering visual element evolutions for historical document dating. In: 2016 15th International conference on frontiers in handwriting recognition (ICFHR). pp. 7–12. IEEE (2016)
13. He, S., Samara, P., Burgers, J., Schomaker, L.: Historical manuscript dating based on temporal pattern codebook. Computer Vision and Image Understanding **152**, 167–175 (2016)
14. He, S., Sammara, P., Burgers, J., Schomaker, L.: Towards style-based dating of historical documents. In: 2014 14th International Conference on Frontiers in Handwriting Recognition. pp. 265–270. IEEE (2014)
15. Hosoe, M., Yamada, T., Kato, K., Yamamoto, K.: Offline text-independent writer identification based on writer-independent model using conditional autoencoder. In: 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 441–446. IEEE (2018)
16. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
17. Keglevic, M., Fiel, S., Sablatnig, R.: Learning features for writer retrieval and identification using triplet cnns. In: 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 211–216. IEEE (2018)
18. Pletschacher, S., Antonacopoulos, A.: The page (page analysis and ground-truth elements) format framework. In: ICPR. pp. 257–260. IEEE (2010)
19. Richler, B.: Hebrew manuscripts in the vatican library: catalogue. Hebrew manuscripts in the Vatican Library pp. 1–790 (2008)
20. Richler, B., Beit-Arié, M.: Hebrew manuscripts in the biblioteca palatina in parma: catalogue; palaeographical and codicological descriptions (2011)
21. Savitzky, A., Golay, M.J.: Smoothing and differentiation of data by simplified least squares procedures. Analytical chemistry **36**(8), 1627–1639 (1964)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
23. Sirat, C.: Hebrew manuscripts of the Middle Ages. Cambridge University Press (2002)

24. Studer, L., Alberti, M., Pondenkandath, V., Goktepe, P., Kolonko, T., Fischer, A., Liwicki, M., Ingold, R.: A comprehensive study of imagenet pre-training for historical document image analysis. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 720–725. IEEE (2019)
25. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
26. Wolf, L., Potikha, L., Dershowitz, N., Shweka, R., Choueka, Y.: Computerized paleography: tools for historical manuscripts. In: 2011 18th IEEE International Conference on Image Processing. pp. 3545–3548. IEEE (2011)
27. Yardeni, A., et al.: The book of Hebrew script: history, palaeography, script styles, calligraphy & design. Carta Jerusalem (1997)