Is a deep learning algorithm effective for the classification of medieval Hebrew scripts?

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

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

In this paper, we are presenting an interdisciplinary project that applies deep learning models to classify script types and sub-types in medieval Hebrew manuscripts. It incorporates the techniques and databases of Hebrew paleography and (with reservations) Hebrew codicology. This research project is part of our ongoing effort to develop algorithmic tools for processing historical documents within the Visual Media Lab at the Department of Computer Science at Ben-Gurion University of the Negev, Israel.[1]

The ongoing digitization of manuscripts' collections kept in different libraries worldwide leads to the increasing availability of more and more volumes of manuscripts that once could have been only studied *in situ*. We have all reasons to believe that, within a few years, thousands of more manuscripts around the globe would be properly digitized and available online. In the case of Hebrew manuscripts, this process is already very advanced, with the Institute for Microfilmed Hebrew Manuscripts at the National Library of Israel that already hosts more than 70,000 microfilms and thousands of digital images. These digitized documents make more than 90% of the known Hebrew manuscripts. Thus, automatic processing, or at least the primary computerized categorization of manuscripts, has become the most urgent task of modern Hebrew paleography.

Hebrew paleography emerged in the mid-20s century, side by side with the modern Latin paleography, and with the same basic principles. The theoretical basis of Hebrew paleography is formulated in the works of Malachi Beit-Arie,[2] Norman Golb,[3] Benjamin Richler[4], Colette Sirat[5], Ada Yardeni[6]. Contemporary Hebrew paleography identifies six main-types of scripts: Ashkenazi, Italian, Sephardic, Oriental, Byzantine, Yemenite. Each main script type may contain up to three sub-types of scripts: square, semi-square, cursive. In total, there are 15 Hebrew script sub-types. The paleographical classification of the

[1] The participation of Dr. Vasyutinsky Shapira in this project is funded by Israeli Ministery of Science, Technology and Space, Yuval Ne'eman scholarship n. 3-16784.

[2] Beit-Arié, Malachi. *Hebrew codicology*. Jerusalem: Israel Academy of Sciences and Humanities, 1981; Beit-Arié, Malachi and Edna Engel, *Specimens of mediaeval Hebrew scripts*, in 3 vol. Israel Academy of Sciences and Humanities, 1987, 2002, 2017.

[3] Golb, Norman, and Omeljan Pritsak. *Khazarian Hebrew documents of the tenth century*. Cornell University Press, 1982.

[4] Richler, Binyamin, and Malachi Beit-Arié, eds. *Hebrew manuscripts in the Biblioteca Palatina in Parma: catalogue*. Jerusalem: Hebrew University of Jerusalem, Jewish National and University Library, 2001; Richler, Benjamin, Malachi Beit-Airé, and Nurit Pasternak. "Hebrew manuscripts in the Vatican Library." *Catalogue. Compiled by the Staff of the Institute of the Microfilmed Hebrew Manuscripts, Jewish National and University Library (Città del Vaticano)*, 2008.

[5] Sirat, Colette. *Hebrew manuscripts of the Middle Ages*. Cambridge University Press, 2002.

[6] Yardeni, Ada. *The book of Hebrew script: history, palaeography, script styles, calligraphy & design*. Carta Jerusalem, 1997.

ground truth for our project comes from the SfarData dataset,[7] which includes full codicological descriptions and paleographical definitions of all dated medieval Hebrew manuscripts until the year 1540 (this makes about 95% of the known dated medieval Hebrew manuscripts). The SfarData project was initiated by Malachi Beit-Arié in the 1970s and it is currently hosted at the site of the National Library of Israel.

Our project is an on-going research. Our current goal is to develop algorithms to recognize Hebrew scripts and their sub-types. The practical applications at this stage would include:

- Determining the date and the area of writing. The paleographical classification of verified manuscripts enables machine learning models to learn the features common to each type and sub-type. The trained models can determine the sub-type of a query manuscript, which enables estimating the date of an undated manuscript or the place of copying. Thus, the application of this technology to fragmentary and faked text has the potential to roughly estimate where and when they were written. Today this task poses serious challenges and often only an experienced librarian or paleographer is capable of a plausible guess. There are many forged and incorrectly dated manuscripts, on the basis of which historical theories and histories of entire peoples are built. The use of a well-trained algorithm will allow us to objectively resolve such issues.
- Already at this stage, we expect the algorithm to be capable of producing a rough catalogue of a collection of manuscripts where no trained human paleographer is available. Alongside the effort of the Institute for Microfilmed Hebrew Manuscripts to assemble the digital images of all the known Hebrew manuscripts, there still are important collections that have not been digitized and properly catalogued, such as the big collection of Hebrew manuscripts in the Vernadsky Library in Kyiv, currently in the most alarming state of preservation. Even the basic catalogue made by the algorithm could attract to such collections the much-needed attention of the researchers.
- Identifying important parts of a manuscript, such as colophons, owner's notes. These additions to a manuscript are often written in a different script sub-type. Identifying them allows a researcher to recognize the date, place of copying, name of the scribe, etc.
- Tracking the movement of scribes, scholars, and communities over time through script and/or hand similarities.
- When the algorithm is further trained to recognize specific words, we would apply it to the biggest manuscripts' collections, such as Firkowicz collections kept in St. Petersburg, collections of Bibliothèque nationale de France, and others. This will allow us to have a closer look at some intriguing and fascinating but extremely complicated objects of research, when pieces of information about them are scattered in the libraries around the globe. To bring just one example, we could learn more about the Jews of Magna Graecia, with their physical and social mobility and intricate history. The relevant manuscripts from different libraries' collections can be identified, brought together, connected, and sorted out with the help of machine learning.
- Another possible application at this stage includes research of little-known, challenging, and often mysterious marginal Jewish communities, such as Georgian, Bukharan, Mountain Jews, about whom little is known today and whose history remains to a great extent legendary. History and works of the Jews of the Kingdom of the Two Sicilies and the Jews of Malta (to whom belonged the famous kabbalist Abraham Abulafia) before the expulsion by the king of Aragon, is another example of a potential application of the algorithm.

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

There are several ongoing projects in the research of the Hebrew manuscripts that complement ours; the most important among them are the Friedberg Genizah Project with its Cairo Genizah site[8] and the Judeo-Arabic corpus[9], the eScriptorium,[10] and the Haifa Project for Research on the Dead Sea Scrolls[11]. There is also a very promising project at the Bar-Ilan university that works on building Hebrew manuscript metadata records and is focused on the manuscripts dated after 1540, i.e., later than the classical Hebrew paleography.[12] Similar efforts to train an algorithm to recognize script types and built a web database exist in Latin paleography,[13] with its database.[14] A recent deep learning method[15] studies the impact of varying patch sizes on the performance of writer identification for modern handwritten documents. Their results expose that the performance depends on the patch size and for each dataset a different patch size gives the best performance. Arabadjis et al.[16] classify the hands who wrote a given set of historical Byzantine Codices using manually designed features for matching a similarity score.

In our project, we built a medieval Hebrew manuscripts dataset, Visual Media Lab - Hebrew Paleography (VML-HP). The VML-HP dataset includes 500 pages labeled with 15 script types. To our best knowledge, this is the first publicly available Hebrew paleographic dataset. Currently, the dataset can be downloaded from https://www.cs.bgu.ac.il/~berat/. To provide a common baseline for algorithms assessment and comparison, we supply the partition of the VML-HP. The dataset is split into training and two test sets. The first test set, the typical test set, consists of unseen pages of documents present in the training set. The second - blind test set - contains unseen manuscripts and imitates a real-life scenario. We present a case study for script type classification on the introduced dataset. We introduce a homogeneous style patch extraction method, where each patch contains a fixed number of lines. We also compare several established deep learning classification models and preprocessing methods. The obtained results show that there is a big room for improvement on the blind test set, whereas the typical test set is an easier problem. Currently, we are working on exploring more advanced deep learning architectures that can capture fine-grained features of the Hebrew manuscripts. The fine-grained features are the features that aim to differentiate between hard-to-distinguish object classes, such as subtle differences in letter forms in different script sub-types.

[8] https://fjms.genizah.org/

[9] http://fjms.genizah.org/

[10] https://www.escriptorium.uk/

[11] http://megillot.haifa.ac.il/index.php/en/

[12] Prebor, Gila, Maayan Zhitomirsky-Geffet, and Yitzchak Miller. "A new analytic framework for prediction of migration patterns and locations of historical manuscripts based on their script types." *Digital Scholarship in the Humanities* 35.2 (2020): 441-458.

[13] Cloppet, Florence, et al. "Icdar2017 competition on the classification of medieval handwritings in latin script." *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Vol. 1. IEEE, 2017: 1371-1376; Studer, Linda, et al. "A comprehensive study of imagenet pre-training for historical document image analysis." *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019: 720-725.

[14] http://www.digipal.eu/

[15] Punjabi, Akshai, et al. "Writer identification using deep neural networks: Impact of patch size and number of patches." 2020 International Conference on Pattern Recognition. IEEE, 2020: 3065-3068.

[16] Arabadjis, Dimitrios, et al. "A general methodology for identifying the writer of codices. Application to the celebrated "twins"." *Journal of Cultural Heritage* 39 (2019): 186-201.

Method

We propose to develop a computational tool that can recognize the script sub-type of a given Hebrew manuscript. Conventional recognition methods utilize handcrafted features, which mainly depend on careful design and expert knowledge. More advanced recognition methods are based on deep learning and can acquire effective feature representations from training data. Deep learning algorithms are backboned by neural networks inspired by human brain architecture, consisting of neurons and synapses among them. A deep learning algorithm is organized as a stack of layers, each of which is a collection of feature extractors, so-called filters (Fig.1). Raster pixel values of a document image patch are fed into the network and are transformed into feature maps as they pass forward through the layers. Each layer extracts features at a different abstraction level. Initial layers detect primitive features such as dots, lines, and curves. Final layers combine these features into complex features such as corners, circles, and letters. At the final layer, the document image patch is classified into one of the script sub-types. The filters are updated according to a measure of the difference between the target and the predicted labels. The major drawback of a deep learning network is the necessity of a large amount of labeled train data, for example, 1000 samples per class.

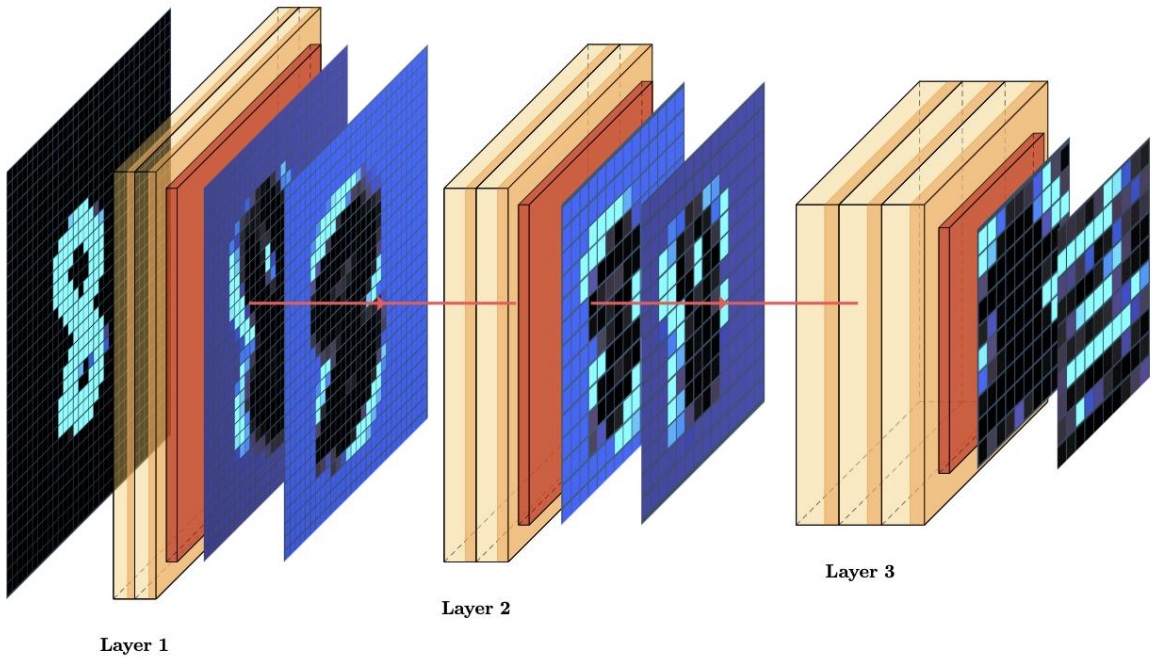


Fig.1: Illustration of a deep learning network. It consists of stacked feature extraction layers. The early layers extract primitive features, and the later layers extract more complex features.

Building the dataset

Sfardata, the database of Hebrew paleography and codicology, completed by Malachi Beit-Arié and his team, contains descriptions and classification of almost all known dated medieval Hebrew manuscripts. All the manuscripts in the database were studied in the libraries where they were kept, and many of them are now days available online. Malachi Beit-Arié and his team met with our team, discussed our project, gave us their full support, and allowed us to use their database in its entirety. Our team's paleographer, who is herself a student of Malachi Beit-Arié, handpicked digitized pages from the manuscripts described

in the SfarData as the raw material for our project. When for certain script sub-types we had to add manuscripts not described in SfarData, our paleographer picked them in accordance with the classification of medieval Hebrew manuscripts as described in SfarData.

Pages in the VML-HP dataset were extracted from high-quality digitized manuscripts, and we gave first preference to those kept in the National Library of Israel. We also used manuscripts from other libraries, first and foremost the British Library and the Bibliothèque nationale de France, with their significant collections of digitized manuscripts available for download.The dataset includes 500 pages total.

Clean patch generation algorithm

The VML-HP dataset contains 500 pages that represent 15 script sub-types. The ideal solution is to feed whole pages into the network because with larger input images, the network can capture more fine-grained features[17]. However, the input image cannot be greater than the size that fits the memory requirements. Therefore, we balance this tradeoff by cropping image patches that contain approximately five text lines, which is a sufficient size for human paleographers to classify the script type. Some parts of the pages contain irrelevant information, such as decorations, marginal drawings, or noisy background, as illustrated in Fig.2. Therefore, we developed a clean patch generation algorithm (https://www.cs.bgu.ac.il/~berat/data/hp_dataset.zip) that generates patches containing pure text regions and an approximately equal number of text lines.

To achieve this, we first calculate a square patch size for each page s×s that will include five lines. Then, we extract random patches of size s×s. The size of the extracted patches, i.e., the value of s, varies across manuscripts. Therefore, to remain consistent with our previous experiments, the patches are resized to 350× 350. Examples of such clean patches are shown in Fig. 3.

Calculating the patch size s×s for each page is done by first, extracting k random patches of the size equal to one-tenth of the page height, as a patch of this size usually includes several text lines. Then The number of lines in a given patch is computed by counting the peaks of the y profile using Savitzky-Golay filter. Finally, the desired patch size is given by $s = \frac{h}{10} \times \frac{n}{m}$, 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. We used $n = 5$ and $k = 20$.

Furthermore, 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:

[17] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." *International Conference on Machine Learning*. PMLR, 2019.

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

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.

Fig.2: Example output patches from a naive patch generation algorithm. Some patches contain irrelevant features, some contain only a few characters, and others do not even contain any text.

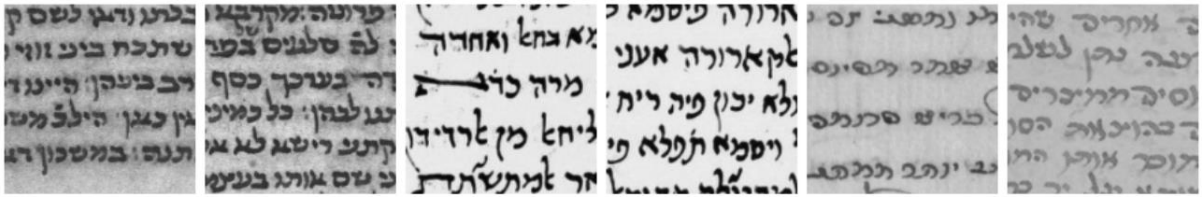


Fig. 3: Example output patches from the clean patch generation algorithm.

Results

We experimented with several convolutional network architectures. In all the experiments, we train the network on the training set and test it on both test sets, the typical and the blind. We generated 150K patches from the training set, 10K patches from the typical test set, and 10K patches from the blind test set. The patches are generated using the clean patch extraction algorithm described in the previous section and are resized to the size of 350×350 pixels. The generated patches are equally distributed amongst all of the script types. The classification results are evaluated by the patch level accuracy and the page level 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.

Classifying into 15 script types

Table 1 shows the accuracy results for classifying 15 script sub-types using different convolutional networks and compares the results on the typical and blind test sets at patch and page levels. As we can see from the results, the typical test set patches and pages are easier to classify. The gap in results on typical and blind test sets shows that the models are overfitting. The models have seen the pages from the typical test set during the training; however, the blind test set contains pages from unseen manuscripts. The difference in results shows that the models' learned features are specific to the manuscripts and not to the script type, like background texture. At nearly all levels and sets, the performance of the ResNet50 classifier is consistently higher; however, it does not surpass 40% accuracy on the blind test set. The random guess accuracy of 15 classes is 7.6%, indicating that the network can extract some script type features and improves the random classification accuracy. We can argue that script type classification is an expressible function, but the network needs more data to learn this function.

	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**
InceptionV3	94.94	26.41	95.89	26.71

Table 1: Patch and page level accuracies on typical test set and blind test set using different network architectures for classifying 15 script sub-types.

Classifying square and cursive script types

Table 2 shows the accuracy results for classifying only two script types, square and cursive, using different convolutional networks. From a human paleographer's point of view, it is almost impossible to make a mistake and mix up square and cursive script (while the boundaries between square and semi-square, and semi-square and cursive can be blurry). Thus, a good result obtained by the algorithm in this case, indicates that the algorithm learns the correct features in the manuscript, that represent the script itself. We can note that the typical test set accuracy is fully saturated, whereas there is still little room for improvement at blind test set accuracy. This result strengthens the above argument that more samples should be used in the training phase; we see that decreasing the number of classes from 15 to two (which increased the number of samples per class), leads to higher accuracy.

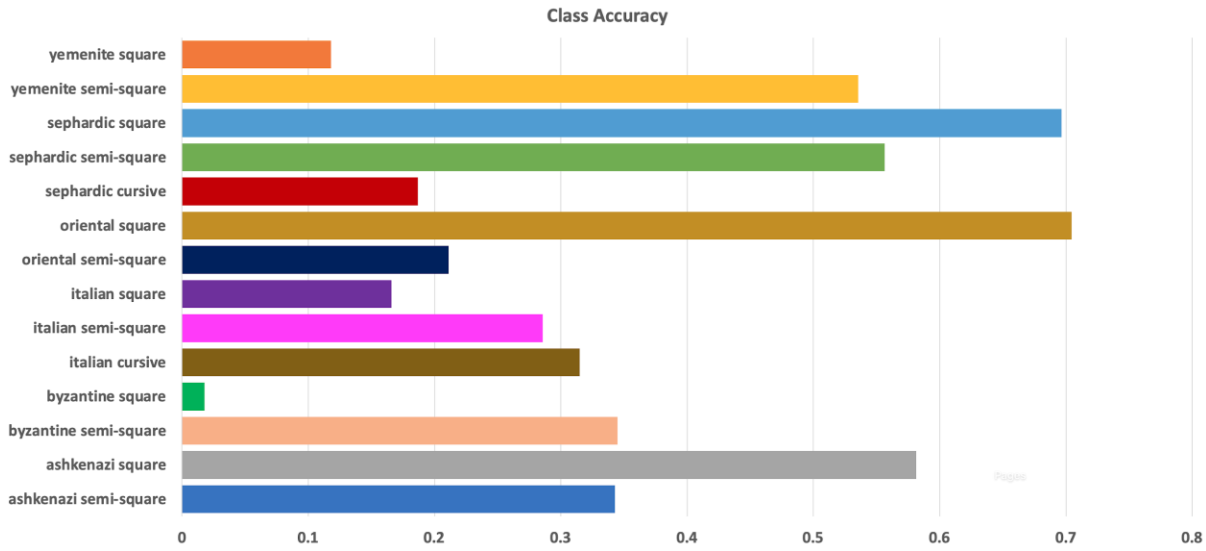
	Patch level		Page level	
	Typical	Blind	Typical	Blind
DenseNet	99.85	87.06	100	83.72
AlexNet	99.48	88.01	100	91.86
VGG11	99.93	86.45	100	88.37
ResNet18	96.65	86.85	100	87.21
ResNet50	99.99	90.58	100	94.19
SqueezeNet	98.03	82.45	100	86.05
InceptionV3	99.16	82.06	100	20.23

Table 2: 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.

Discussion

The classification accuracy of the best performing model, i.e. ResNet50, is around 35%. When comparing class accuracies (Fig. 4) we found that particular classes have an accuracy over 50%, i.e., Yemenite semi-square, Sephardic square, Sephardic semi-square, Oriental square, and Ashkenazi semi-square. These results indicate that a classification system designed only for these classes will have higher page level accuracy, since the page level accuracy is computed by the majority vote over the patches from the same page.

Fig. 4: Patch level class accuracies on blind test set using ResNet50 network.



The Byzantine square sub-type has a very low accuracy because it was confused with Byzantine semi-square (Fig. 5). Interestingly, this confusion is not mutual because the Byzantine semi-square sub-type was confused with Italian. In contrast, the confusion among the Italian, Oriental, Sephardic and Yemenite semi-square sub-types are mutual. The mutual confusions can be due to the paleographers' ambiguity in the ground truth of semi-square types or to insufficient ground truth.

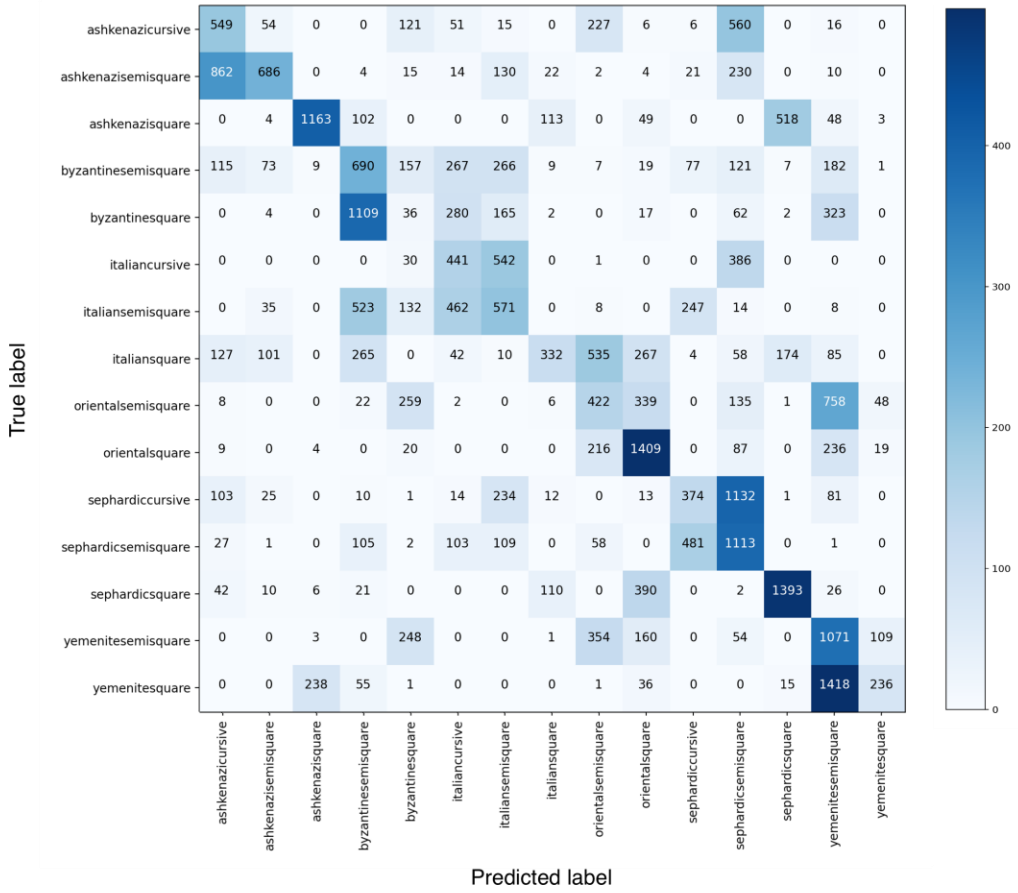


Fig. 5: Patch level confusion matrix on blind test set using ResNet50 network.

Conclusions

From the paleographic point of view, it would be beneficial to gain insight into the features that underlie the class decisions. We are developing a fine-grained classification model that can spot the regions taken into account for script type decisions. In addition, we are collecting and labeling more document page images, as the machine learning for a 15-class problem requires around 15K samples in total. Our algorithm significantly surpasses the random guess accuracy of 15 classes (7.6%) and this indicates that the network can extract some script type features. We can argue that script type classification is an expressible function, but the network needs more data to learn this function. When more material is brought for comparison and the size of the test, train, and blind sets increases, the accuracy of the algorithm will improve.

Our work and its place in the overall theme of Jewish Studies in the Digital Age

Our project belongs to the field of digital research of manuscripts and historical documents. The amount of digitally available manuscripts and documents in different libraries and archives is constantly growing and already the amount of material available is often more than an individual researcher could process manually. In all likelihood, in the foreseeable

future, a human researcher will formulate a problem and the processing of large amounts of data will be assigned to an algorithm. For this to be possible, algorithms must recognize, classify, and ultimately search through large amounts of unrecognized manuscripts and documents. The integration of computer-based techniques can now bring to the manuscripts' research the often-missing quality of objectivity, possibility of objective verification of results. It also brings with it the possibility of solving problems that are beyond the physical capacities of an individual researcher.

We use the theoretical framework of Hebrew paleography to train deep learning neural networks to classify Hebrew script types and sub-types and our project works side by side and compliments such ongoing project as eScriptorium, Friedberg Genizah Project and more.