



## Cuneiform Stroke Recognition and Vectorization in 2D Images


Adéla Hamplová <hamplova\_at\_pef\_dot\_czu\_dot\_cz>, Czech University of Life Sciences Prague  <https://orcid.org/0000-0002-1012-650X>

Avital Romach <avital\_dot\_romach\_at\_yale\_dot\_edu>, Yale University  <https://orcid.org/0000-0001-9199-3228>

Josef Pavlíček <pavlicek\_at\_pef\_dot\_czu\_dot\_cz>, Czech University of Life Sciences Prague  <https://orcid.org/0000-0002-3959-5406>

Arnošt Veselý <vesely\_at\_pef\_dot\_czu\_dot\_cz>, Czech University of Life Sciences Prague  <https://orcid.org/0000-0001-8979-1336>

Martin Čejka <cejkamartin\_at\_pef\_dot\_czu\_dot\_cz>, Czech University of Life Sciences Prague  <https://orcid.org/0000-0002-2909-486X>

David Franc <francd\_at\_pef\_dot\_czu\_dot\_cz>, Czech University of Life Sciences Prague  <https://orcid.org/0000-0003-3160-9559>

Shai Gordin <shaigo\_at\_ariel\_dot\_ac\_dot\_il>, Ariel University; Open University of Israel  <https://orcid.org/0000-0002-8359-382X>

### Abstract

A vital part of the publication process of ancient cuneiform tablets is creating hand-copies, which are 2D line art representations of the 3D cuneiform clay tablets, created manually by scholars. This research provides an innovative method using Convolutional Neural Networks (CNNs) to identify strokes, the constituent parts of cuneiform characters, and display them as vectors — semi-automatically creating cuneiform hand-copies. This is a major step in optical character recognition (OCR) for cuneiform texts, which would contribute significantly to their digitization and create efficient tools for dealing with the unique challenges of Mesopotamian cultural heritage. Our research has resulted in the successful identification of horizontal strokes in 2D images of cuneiform tablets, some of them from very different periods, separated by hundreds of years from each other. With the Detecto algorithm, we achieved an F-measure of 81.7% and an accuracy of 90.5%. The data and code of the project are available on GitHub.

## 1 Introduction

### 1.1 Cuneiform Texts and Artificial Intelligence

Cuneiform is one of the earliest attested writing systems, and for thousands of years cuneiform has also been the dominant script of the ancient Middle East, a region stretching roughly from the Persian Gulf to modern Turkey's highlands, and south across the Levant into Egypt. Cuneiform texts appeared from the end of the fourth millennium BCE until they fell out of use in the early centuries CE. The script was used for writing a plethora of different documents: legal, administrative, and economic documents; correspondence between private individuals, or high-officials and their kings; some of the oldest works of literature; royal inscriptions describing the deeds of great kings; as well as lexical and scientific compendia, some of which form the basis of the Greco-Roman sciences the Western world is built upon today. Hundreds of thousands of cuneiform documents have been discovered since excavations began in the 1850s. Recent estimates indicate the cuneiform text corpus is second in size only to that of ancient Greek [Streck 2010].

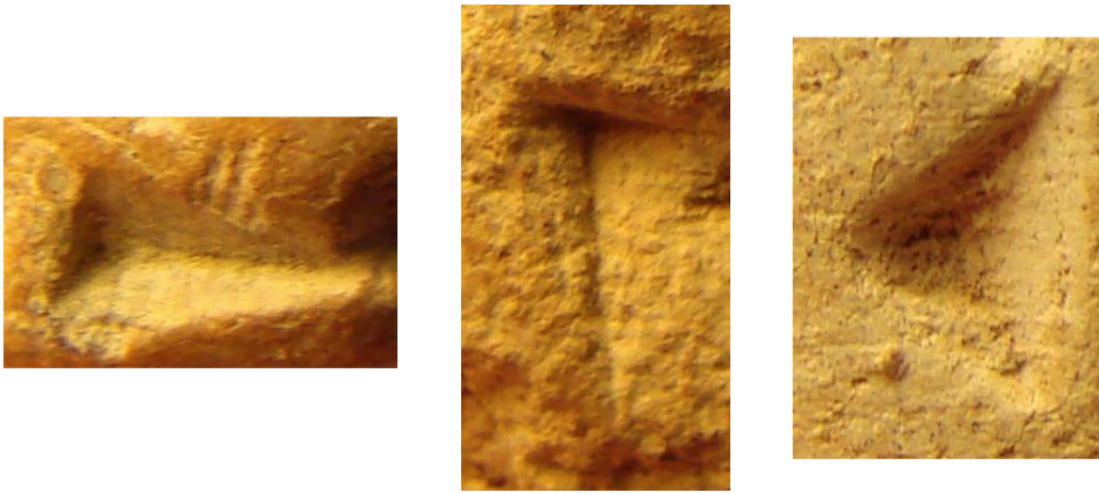
The rising application of artificial intelligence to various tasks provides a prime opportunity for training object detection models to assist the digital publication of cuneiform texts on a large scale. This will help set up a framework for cultural heritage efforts of preservation and knowledge dissemination that can support the small group of specialists in the field.

Cuneiform provides unique challenges for object detection algorithms, particularly OCR methods. Cuneiform tablets, on which the texts were written, are 3D objects: pieces of clay which were shaped to particular sizes. While the clay was still moist, scribes used styli with triangular edges to create impressions on the clay in three possible directions: horizontal, vertical, or oblique (also *Winkelhaken*; see Figure 1). “Diagonal” strokes are also found in the literature, but these are technically either another type of horizontal or an elongated oblique impression [Cammarosano 2014] [Cammarosano et al. 2014] [Bramanti 2015]. Each of these impressions is called a stroke (or wedge, due to their shape). Combinations of different strokes create characters, usually referred to as signs [Taylor 2015]. Cuneiform can be more easily read when there is direct light on the tablet, especially from a specific angle that casts shadows on the different strokes.

1

2

3



**Figure 1.** The main cuneiform strokes taken from Neo-Babylonian signs, from left to right: AŠ (<https://labasi.acdh.oew.ac.at/tablets/glyph/detail/10341>), DIŠ (<https://labasi.acdh.oew.ac.at/tablets/glyph/detail/10474>), and U or Winkelhaken (<https://labasi.acdh.oew.ac.at/tablets/glyph/detail/11430>), as recorded in the LaBaSi palaeographical database.

From the inception of research in cuneiform studies, tablets were difficult to represent in a modern 2D publishing format. Two solutions were found: 4

- When possible, 2D images of cuneiform tablets were taken. However, for most of the field's history, such images were extremely costly to produce and print, and they were often not in sufficient quality for easy sign identification (for the history of early photography and cuneiform studies, see [Brusius 2015]).
- The second solution was creating hand-copies, 2D black and white line art made by scholars of the tablets' strokes. This was the most popular solution. The disadvantage of this method is that it adds a layer of subjectivity, based on what the scholar has seen and on their steady drawing hand. Nowadays these hand-copies are still in use, often drawn using vectors in special programs (the most popular being the open source vector graphics editor Inkscape).

In recent years, the quality of 2D images has risen significantly, while the costs of production and reproduction dropped. A constantly growing number of images of cuneiform tablets are currently available in various online databases. The largest repositories are the British Museum, the Louvre Museum, the Cuneiform Digital Library Initiative, the Electronic Babylonian Library, and the Yale Babylonian Collection. 2D+ and 3D models of cuneiform tablets have also become a possibility, although these are still more expensive and labour-intensive to produce [Earl et al. 2011] [Hameeuw and Willems 2011] [Collins et al. 2019]; cf. overview in [Dahl, Hameeuw, and Wagensohn 2019]. 5

Previous research of identifying cuneiform signs or strokes have used mostly 3D models. Two research groups have developed programs for manipulating 3D models of cuneiform tablets: the CuneiformAnalyser [Fisseler et al. 2013] [Rothacker et al. 2015] and GigaMesh [Mara et al. 2010]. Each group developed stroke extraction through geometrical features identification, while one team also used the 3D models for joining broken tablet fragments [Fisseler et al. 2014]. In addition, the GigaMesh team extracted strokes as Scalable Vector Graphic (SVG) images [Mara and Krömker 2013], which practically means creating hand-copies automatically as vector images. This was used as a basis for querying stroke configurations when looking for different examples of the same sign, using graph similarity methods [Bogacz, Gertz, and Mara 2015a] [Bogacz, Gertz, and Mara 2015b] [Bogacz, Howe, and Mara 2016] [Bogacz and Mara 2018] [Kriege et al. 2018]. 6

Work on hand-copies includes transforming raster images of hand-copies into vector images (SVG) [Massa et al. 2016]. Hand-copies and 2D projections of 3D models were used for querying signs by example using CNNs with data augmentation [Rusakov et al. 2019]. Previous work on 2D images has only recently started. Dencker et al. used 2D images for training a weakly supervised machine learning model in the task of sign detection in a given image [Dencker et al. 2020]. Rusakov et al. used 2D images of cuneiform tablets for querying cuneiform signs by example and by schematic expressions representing the stroke combinations [Rusakov et al. 2020]. A more comprehensive survey of computational methods in use for visual cuneiform research can be found in [Bogacz and Mara 2022]. 7

As there are no published attempts to extract strokes directly from 2D images of cuneiform tablets, the purpose of this paper is a proof of concept to show it is possible to extract and vectorize strokes from 2D images of cuneiform using machine learning methods. The quantity and quality of 2D images is improving, and for the most part they provide a more accurate representation of the tablet than hand-copies, as well as being cheaper and quicker to produce in comparison to 3D models. Furthermore, since there are only three basic types of strokes, but hundreds of signs and variants, one can label a significantly smaller number of tablets to attain a sufficient number of strokes for training machine learning models. The resulting model will be able to recognize strokes in cuneiform signs from very different periods, separated by hundreds of years. This semi-automation of hand-copies will be a significant step in the publication of cuneiform texts and knowledge distribution of the history and culture of the ancient Near East. Our data and code are available on GitHub.<sup>[1]</sup> 8

## 1.2 Object Detection

Identifying cuneiform signs or strokes in an image is considered an object detection task in computer vision. Object detection involves the automatic 9

identification and localization of multiple objects within an image or a video frame. Unlike simpler tasks such as image classification, where the goal is to assign a single label to an entire image, object detection aims to provide more detailed information by detecting and delineating the boundaries of individual objects present. Object detection algorithms work by analysing the contents of an image and searching for specific patterns or features that are associated with the object. These patterns or features can include things like color, texture, shape, and size.

There are different types of computational models for object detection, ranging from the purely mathematical to deep learning models [Wevers and Smits 2019]. *Mathematical models* often involve traditional computer vision techniques that rely on well-defined algorithms and handcrafted features. These methods typically follow a series of steps to detect objects in an image. First is extracting relevant features from the image (edges, corners, textures, or color information), where the algorithms are usually adapted based on domain knowledge. Then mathematical operations are performed to determine the location and extent of potential objects based on the identified features. Further methods can be used in post-processing to refine the results. Computational methods work best in ideal or near-ideal conditions, meaning there needs to be standardization in the types of cameras used for taking the images, the lighting situation, and the background. The objects themselves should also be as uniform as possible in size, shape, and color. This means that in complex and diverse real-world scenarios, mathematical models are often insufficient for satisfactory results.

*Deep learning models*, particularly convolutional neural networks (CNNs), have revolutionized object detection [Girshick et al. 2014]. Instead of manual feature engineering used by mathematical models, deep learning methods can automatically detect relevant features and objects. The models are trained on labelled data: a set of images where the objects of interest have been marked, usually in rectangular bounding boxes, by humans. After training, the models can be tested and used on unseen images that were not in the labelled training dataset to detect the same type of objects. Their biggest advantage is that they can handle a wide range of object shapes, sizes, and orientations, making them adaptable to diverse scenarios, and generalize well across different datasets. The disadvantages of such models are that they require large amounts of labelled data for effective training; they are more computationally intensive compared to traditional mathematical models, requiring more computational power outside the scope of the average computer; and they are black boxes, meaning it is not always possible to explain why the model makes a certain prediction or not.

In a previous research project, we combined the use of mathematical and deep learning object detection methods for wildlife mapping [Pavliček 2018]. However, for this project, mathematical models proved insufficient for the complexity and variability of images of cuneiform tablets, which include different tablet shapes, colors, broken sections, etc. Therefore, in this article we present our results on stroke recognition for 2D images of cuneiform tablets using several deep learning models, and compare their advantages and disadvantages for this type of object detection on ancient and complex writing systems.

### 1.2.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) are multilayer networks, specifically designed for processing and analyzing visual data. The convolutional layers in the network process the input image through different filters that help the network detect features of interest like edges, corners, textures, etc. The additional layers process the resulting feature maps to detect relevant combinations of features. There can be several iterations of convolutional layers, depending on the specific architecture of the neural network used.

There are two main types of convolutional neural networks: two-stage detectors and single-stage detectors [Jiao et al. 2019]. Two-stage detectors, like Faster R-CNN (Region-based Convolutional Neural Network), first identify regions of the image that might contain objects before analyzing those regions more closely to detect objects [Girshick et al. 2014]. Single-stage detectors, like YOLO (You Only Look Once; [Redmon et al. 2016]), can detect objects directly without first identifying regions of interest. In what follows, we provide a brief overview of the advantages and disadvantages of both methods. See also [Mishra 2022].

#### 1.2.1.1 YOLO

YOLO, short for You Only Look Once, is a family of convolutional neural network architectures that was first introduced in 2015 by Joseph Redmon et al [Redmon et al. 2016]. The version used in this paper, YOLOv5, was published in 2020 [Jocher et al. 2020]. The YOLOv5 architecture consists of 232 layers<sup>[2]</sup>, multiple convolutional layers that extract features from the image at different scales, and a series of prediction layers, which output the object detection results.

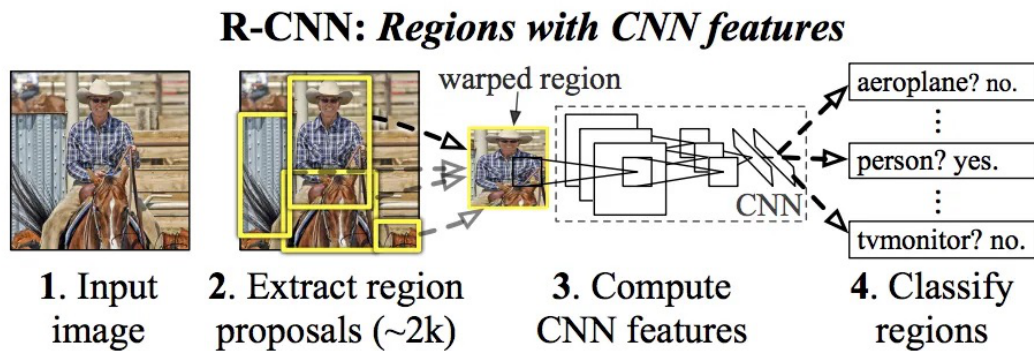
The algorithm divides the input image into a grid of cells, with each cell responsible for predicting the presence of one or more objects. For each cell, the network predicts the confidence score, which reflects the likelihood that an object is present, and the bounding box coordinates that describe the location and size of the object.

The YOLOv5 algorithm has achieved state-of-the-art performance on several benchmark datasets. Its main advantage has been speed: YOLO performs significantly faster than other CNN models. It has become a popular choice for a wide range of applications in computer vision, including object detection in real-time video streams, autonomous driving, and surveillance systems. The latest version at the time of publication is YOLOv8.<sup>[3]</sup>

#### 1.2.1.2 R-CNN, Fast R-CNN, and Faster R-CNN

Region-based Convolutional Neural Networks (R-CNN) were first introduced in 2014 by Ross Girshick et al [Girshick et al. 2014]. This type of detector has four key components: (1) it generates region proposals that suggest potential object locations in an image using a selective search method; (2) it extracts fixed-length feature vectors from each of these proposed regions; (3) for each region of interest, it computes relevant features

for object identification; and (4) based on the extracted CNN features, the regions of interest are classified (see Figure 2).



**Figure 2.** The components of the R-CNN detector, from [Girshick et al. 2014].

A year later, Girshick proposed a more efficient version called Fast R-CNN [Girshick 2015]. In contrast to R-CNN, which processes individual region suggestions separately through the CNN, Fast R-CNN computes features for the entire input image at once. This significantly speeds up the process and allows for better use of storage space for feature storage.

19

Building on the improvements of Fast R-CNN, Faster R-CNN was introduced just three months later [Ren et al. 2015]. It introduced the region proposal network (RPN), which generates regions of interest (RoI), potential identifications of the desired objects. It does so by sliding a small window (known as an anchor) over the feature maps and predicting whether the anchor contains an object or not. For this project, we used a combination of Faster R-CNN and ResNet-50, a deep learning architecture that optimizes the network's performance. This model was implemented by Bi in Detecto python library, using the PyTorch python framework.<sup>[4]</sup>

20

## 2 Dataset and Evaluations

### 2.1 Dataset

#### 2.1.1. Dataset Creation, Division, and Augmentation

The Assyriologists on our team tagged thousands of horizontal strokes in eight tablets from the Yale Babylonian Collection (see Table 1), made available through the kind permission of Agnete W. Lassen and Klaus Wagnonner. For the first stage of research, we labelled 7,355 horizontal strokes in the tablets chosen, divided into 823 images.

21

Yale ID	CDLI ID	Material	Period	Genre	Content	Hand-Copy Publication
YPM BC 014442	P504832	Clay	Neo-Assyrian (ca. 911-612 BCE)	Literary	Enuma Eliš II. 1-16, 143-61	CT 13 1,3
YPM BC 023856	P293426	Clay	Old-Babylonian (ca. 1900-1600 BCE)	Literary	Gilgamesh and Huwawa II. 1-36	JCS 1 22-23
YPM BC 002575	P297024	Clay	Neo/Late-Babylonian (ca. 626-63 BCE)	Commentary	Iqqr Ipuš i 36, ii 31, iii 22, iv 5, v 13	BRM 4 24
YPM BC 016773	P293444	Limestone	Early Old-Babylonian (ca. 2000-1900 BCE)	Inscription	Building inscription of Anam, No. 4	YOS 1 36
YPM BC 016780	P293445	Limestone	Early Old-Babylonian (ca. 2000-1900 BCE)	Inscription	Building inscription of Anam, No. 2	YOS 1 35
YPM BC 016869	P429204	Clay	Middle Assyrian (ca. 1400-1000 BCE)	Inscription	Inscription of Aššur-nadin-apli	YOS 9 71
YPM BC 021204	P308129	Clay	Middle Assyrian? (ca. 1400-1000 BCE)	Medical Text		FS Sachs 18, no. 16
YPM BC 021234	P308150	Clay	Old-Babylonian (ca. 1900-1600 BCE)	Hymn	Hymn to Inanna-nin-me-šar2-ra, II. 52-102	YNER 3 6-7

**Table 1.** Table showing the eight tablets that were labelled and their metadata. The information is taken from the Yale Babylonian Collection website. "CDLI ID" refers to the catalogue number in the Cuneiform Digital Library Initiative database. The publication abbreviations in the column labelled "Hand-Copy Publication" follow the Reallexikon der Assyriologie online list.

To train an artificial neural network, a dataset divided into training, validation, and test subsets needs to be created. In order to increase the number of images in the dataset, several augmentation methods were used. The recommended number of images for each class is at least a thousand images [Cho et al. 2016]. Roboflow<sup>[5]</sup> is a web application used to create extended datasets from manually labelled data using labelling tools such

22

as Labellmg.<sup>[6]</sup>

For pre-processing, the images were divided into equal squares of 416 x 416 pixels (for Detecto and YOLOv5). For R-CNN, the size of images was downsized to 224 x 224 pixels. For the final version of the model, the images were augmented using Roboflow. The final dataset contains 1,975 images with more than 20,000 labels. The augmented horizontal stroke dataset is available online.<sup>[7]</sup>

23

### 2.1.2 Labelling Criteria

For machine learning purposes, the images of the cuneiform tablets needed to be split into squares of equal size (see Appendix). Labelling was performed after splitting the images. This meant the loss of a lot of the context necessary to identify strokes with certainty. We used tablet images with existing hand-copies, which were used as a guide and previous interpretations of the tablets.

24

However, a greater emphasis was given to what is currently visible on the image than what appears in the hand-copy. The hand-copies were not always true to what is seen on the images for three main reasons: (1) the hand-copy preserves signs which were visible on the tablet at the moment of their creation, but by the time the image was taken, they had eroded; (2) the camera angle when taking the image did not capture all the detail the tablet contains. This is a common scenario, since signs at the edges of the tablet will not be seen as clearly when taking a frontal image; and (3) strokes may have been cut off where the image was split.

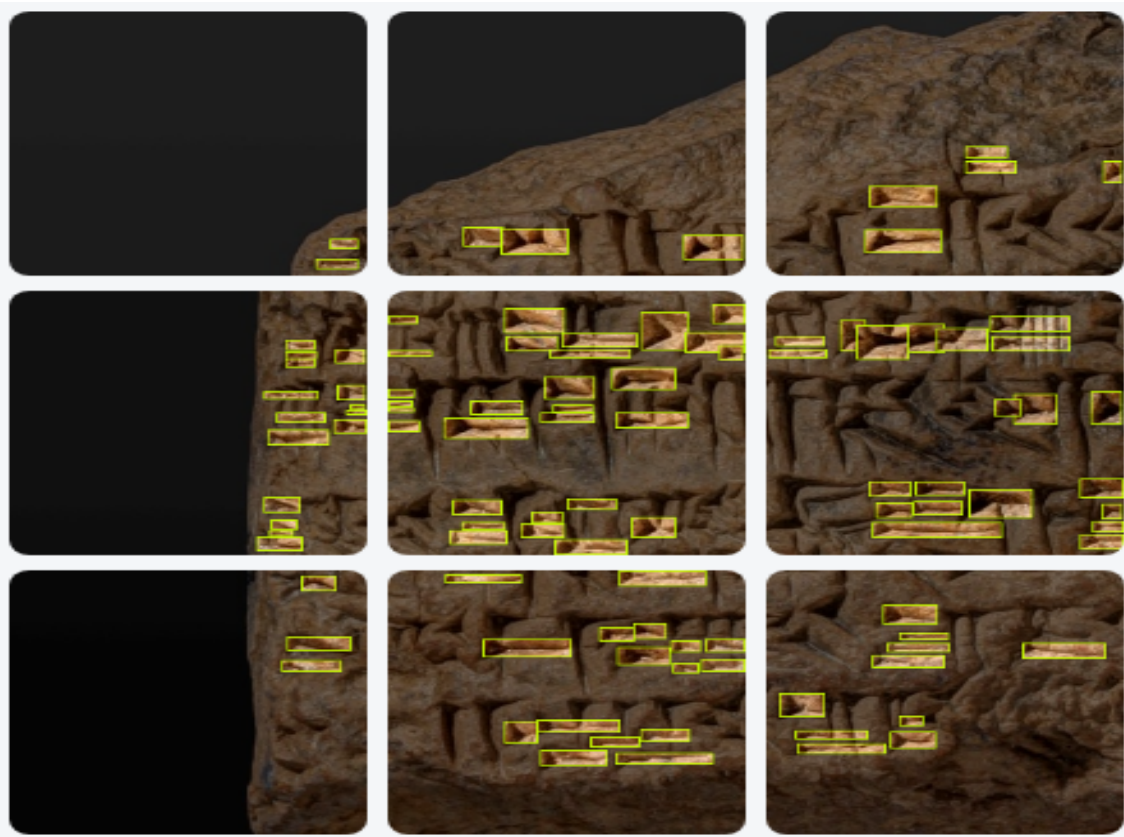
25

If a stroke was completely unrecognizable as a horizontal stroke on the image at hand, because of either of the aforementioned restrictions, it was not labelled. If enough of the characteristic features of the stroke (particularly its triangular head) were present on the image, it was labelled. Being able to identify partially broken strokes is still useful for real-life scenarios, since the tablets themselves are often broken, a common problem in sign identification.

26

Additionally, strokes which are usually considered diagonal were also labelled. A relative leniency was given to this issue, since in general, the lines on cuneiform tablets are not always straight (i.e., creating a 90° angle with the tablet itself). Therefore, a horizontal stroke that may be exactly horizontal when viewed in its line will appear diagonal on the image if the lines themselves are somewhat diagonal. For an example of labelled images, see Figure 3.

27



28

Figure 3. Training set example of labelled horizontal strokes on tablet YPM BC 014442.

## 2.2 Evaluation Metrics

Standard evaluation metrics include precision, sensitivity, and F-measure, calculated from true positive rate (TP), false positive rate (FP), and false negative rate (FN). These are displayed in Table 2. From these, the following metrics can be calculated to quantitatively assess the efficacy of the model.

28

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

**Table 2.** "TP" refers to the the proportion of cases that are correctly identified as positive by the model. "FP" marks the proportion of cases that are incorrectly classified as positive by the model. "FN" reflects the proportion of cases that are incorrectly identified as negative by the model. "TN" indicates the proportion of cases that are correctly identified as negative.

Accuracy or Precision (p): Precision measures how many of the total number of predicted positive cases are true positives. In other words, it is the ratio of true positives to the total number of predicted positive cases, whether true or false.

29

$$p = \frac{TP}{TP + FP}$$

Sensitivity or Recall (s): Sensitivity measures how many of the true positive cases are correctly identified as positive by the model. In other words, it is the ratio of true positives to the total number of positive cases, which includes both true positives and false negatives.

30

$$s = \frac{TP}{TP + FN}$$

F-measure (F1): The F1 measure combines precision and sensitivity into a single score that is commonly used as the final assessment of a model. It is the harmonic mean of precision and sensitivity, calculated as two times the product of precision and sensitivity divided by their sum, resulting in a number between 0 and 1 that can be viewed as a percentage of overall accuracy.

31

$$F = \frac{2 \times p \times s}{p + s}$$

### 3 Results

Our goal was to test several types of object detectors to compare which one gives the best results for our task. According to theoretical comparisons on public datasets, one-stage algorithms (here YOLOv5) should give faster but less accurate results compared to two-stage detectors (here Detecto and R-CNN).

32

While testing with the YOLOv5 detector took only 1.189 seconds, the overall accuracy was just over 40%, which is not sufficient for practical usability. The prediction using the R-CNN network took on average 45 seconds, but the results did not even reach the YOLOv5 level. We believe that this was due to a lack of tuning of the hyperparameters and may be the subject of further experiments. Detecto, which was not as fast as YOLOv5 but not as slow as R-CNN, achieved results that far outperformed both previous algorithms with its 90.5% sensitivity and 81.7% F-score. The reason behind this fact may be that Detecto is an optimised network that combines the principles of a two-stage detector with ResNet. Detailed evaluation results are shown in Table 3, Figure 4, and Table 4.

33

Name	TP	FN	FP	p	s	F1	Fake of All Found Strokes
<b>Detecto (Threshold 0.4)</b>	669	70	229	74.4%	90.5%	81.7%	25.5%
<b>YOLOv5 (Threshold 0.2)</b>	323	419	444	42.1%	43.5%	42.8%	57.8%
<b>YOLOv5 (Threshold 0.3)</b>	256	486	305	45.6%	34.5%	39.2%	54.4%
<b>YOLOv5 (Threshold 0.4)</b>	196	546	190	50.7%	26.4%	34.7%	49.2%
<b>R-CNN (Threshold 0.4)</b>	191	595	941	16.9%	24.8%	19.9%	83.1%

**Table 3.** Evaluation results.

## Evaluation results of object detection algorithms

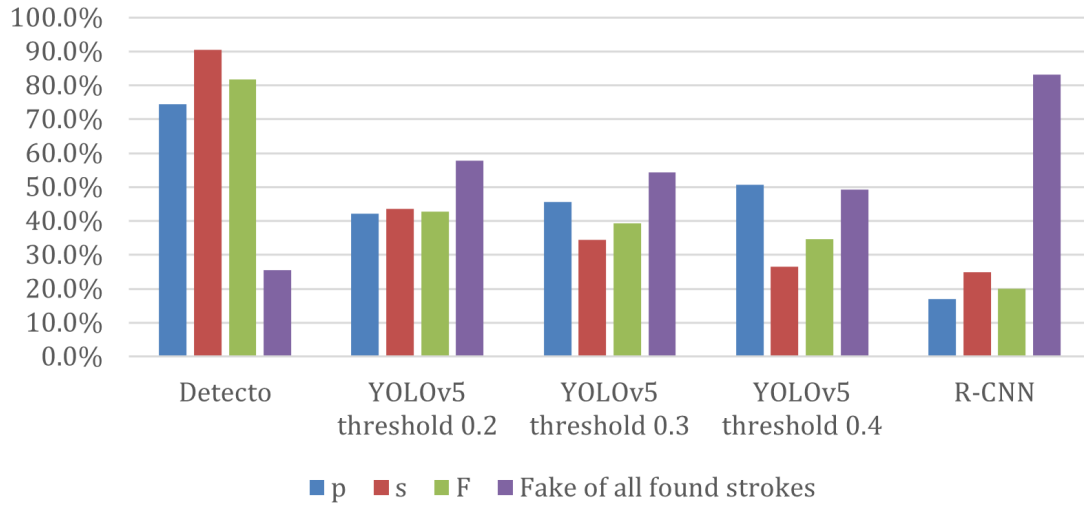


Figure 4. Evaluation results of object detection algorithms.

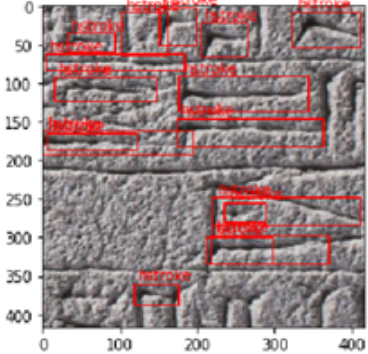
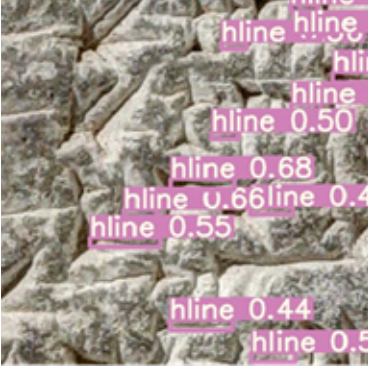

Prediction	Network	Tablet
	Detecto	YPM BC 016773
	YOLOv5	YPM BC 018686
	R-CNN	YPM BC 023856

Table 4. Prediction with respective detectors.

## 4 Discussion

The results of the machine learning model we developed (90.5% accuracy and 81.7% F-measure on Detecto) are very promising, particularly considering the relatively small amount of labelled data. It shows that this previously untested approach, namely stroke identification from 2D images, can be highly efficient for the vectorization of cuneiform tablets. While stroke identification has already been achieved in 3D models of cuneiform-bearing objects (see Section 1.1), our approach shows the same is possible for 2D images which are far cheaper to produce.

Furthermore, our model is not period-dependent, meaning that some of the tablets we have chosen are over a thousand years apart (see Table 1). But since the writing technique itself has not changed during that period, there were no significant differences in the model's ability to recognize the strokes. The major attested difference between stroke types in Mesopotamia is Babylonian vs. Assyrian strokes, the former having longer bodies (the tracing line), and the latter having bigger heads (the central point of the impression left on the clay tablet [Edzard 1980] [Labat 1988]). This, however, does not seem to affect our model.

Since it was possible to effectively identify horizontal strokes, the same is possible for verticals and obliques. With this additional ability, it will be possible to create a full vectorization of cuneiform tablets, which will need to be minimally corrected by an expert. These vectorizations are in effect like hand-copies, which are a first step in assyriological research in interpreting cuneiform texts. It will be the first step in a human-in-the-loop pipeline of cuneiform identification from image to digital text.

The subsequent step, identification of constellations of strokes as cuneiform signs, is under development by part of the authors [Gordin and Romach 2022], currently using traditional hand-copies as input (see demo; see [Yamauchi et al. 2018] for previous work in this direction). Once the signs are identified with their equivalent in Unicode cuneiform [Cohen et al. 2004], these can be transliterated and segmented into words using the model Akkademia, previously developed by the assyriologists on our team and others [Gordin et al. 2020]. This can be further followed by machine

34

35

36

37



translation for the two main languages which used the cuneiform writing system, Sumerian and Akkadian. The Machine Translation and Automated Analysis of Cuneiform Languages (MTAAC) project has begun developing models for translating Ur III administrative texts (dated to the 21st century BCE) [Punia et al. 2020]. Machine translation of Akkadian has also been achieved, focusing primarily on first millennium BCE texts from a variety of genres, available through ORACC [Gutherz et al. 2023].

This pipeline can become a vital part of Assyriological research by making accessible to experts and laypeople alike countless cuneiform texts that have previously received less scholarly attention. However, it is important to note the limitations of this pipeline for Assyriological research.

The vector images we produce are not an accurate representation of the stroke, but rather a chosen schema. Although various schemas can be selected, they are still one simplistic representation on how a stroke looks, which is then applied across the corpus. Therefore, for purposes of scribal hand identification, as well as palaeography, they lack important aspects, such as *ductus*, or the formation of individual signs, and *aspect* (cf. Latin *equilibrium*), or the visual impression created by the set hand of a scribe (i.e., the style of writing). This is the same, however, for manually created hand-copies, since there are limitations to how these 3D objects can be captured in 2D, and some scholars tend to simplify what they see on the tablet when creating hand-copies.

In addition, our results worked well on very high quality 2D images, curated by an expert [Wagensonner 2015]. Although anyone can take high-quality images on their phone, ensuring that the signs and strokes are as legible as possible usually requires an expert knowledge of the cuneiform script and the application of light sources. For this to efficiently work on a large scale, preferably only high-quality 2D images of cuneiform artifacts should be used.

## 5 Towards Quantitative Epigraphy

The task of the epigrapher is to decipher the ancient writing surface — not merely to decipher the script or any linguistic element on its own, but rather to produce a holistic decipherment of the inscription, its material aspects, and its contextual meaning. Therefore, it is challenging to translate epigraphic tasks into one or more computational tasks. The current contribution is a step in this direction, by attempting to gouge out the atomized elements of the script and its arrangement on the writing surface. This diplomatic approach to texts has a long history in medieval scholarly practice [Duranti 1998] [Bertrand 2010], and it is a *desideratum* in order to piece together computational tasks for quantitative epigraphy. It is further a way to bridge the differences across large numbers of ancient or little-known languages and scripts, since much of the literature surrounding their study involves discussions on reconstructing the writing surface, traces, and their proper sequence in their *Sitz im Leben*.

The problem begins when one tries to harmonize tasks from the disciplines in the realm of computer science and the different research questions in the humanities, which do not necessarily overlap. For an epigrapher, identifying and classifying an object in an image is not the end goal, as it might be in computer science. Rather, it is a step in a process to reach historical understanding of a certain genre of text, writing tradition, or historical period. Furthermore, the amounts of data that are available to train generative models like ChatGPT or the many image generator applications made available in recent months, is beyond the scope of the digital data at the hands of the average epigrapher, historian, or digital humanist.

For that end, an interdisciplinary group of scholars dealing with ancient language processing and machine learning for ancient languages [Anderson et al. 2023] [Sommerschild et al. 2023] has set out to better define and standardize data formats, tasks, and benchmarks. This initiative adds to the growing movement of computational and quantitative studies in classics, biblical studies, ancient Near Eastern studies, and so on. The present paper is aimed to contribute to the standardization of the epigrapher's computational tasks in ancient scripts. This paper also provides an example of harmony and collaboration between computer scientists and humanists, as well as between computer science tasks and humanistic research questions.

Furthermore, the methodology and techniques used in this study can be applied to other writing systems beyond cuneiform. The semi-automatic vectorization approach can be adapted to identify and extract specific features of other ancient scripts. In ancient Chinese and Japanese for example, one can try to find the common denominator made up of strokes, the components of each character. In classical Maya writing, one could focus on the anthropomorphic elements of signs, like noses, eyes, ears, etc. The same can be said for other hieroglyphic scripts, like Egyptian or Anatolian hieroglyphs.

## Appendix

### 6.1 Training Convolutional Neural Networks

In the following section, we present the parameters necessary to replicate our results.

For all the models we employed, image augmentation methods were necessary to increase the number of available images for training. Grayscale augmentations were applied with three samples per augmentation:

- Saturation applied to 50% of the images
- Saturation between -20% and +20%
- Exposure between -20% and +20%

#### 6.1.1 Detecto Training

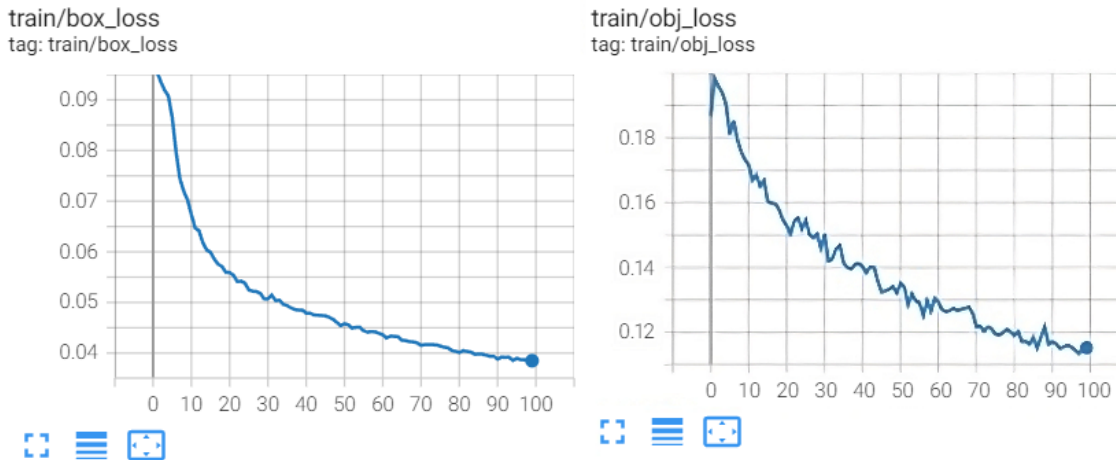
For Detecto training, the dataset was divided into three subsets. The training set contains 3,456 images, the validation set contains 330 images, and the testing set contains 164 images. The training was performed on Google Collaboratory, and the layers from the `fasterrcnn_resnet50_fpn_coco`

258fb6c6.pth model were unfrozen and re-trained. Fifty epochs were run, with three steps per epoch, and the validation loss dropped from 0.74 to 0.64 after all epochs, which took 302 minutes. After five epochs, the validation loss did not decrease, so we could have used early stopping for this model.

### 6.1.2 YOLOv5 Training

The YOLOv5 architecture (with 232 layers) was trained in Google Colaboratory using CUDA on a Tesla T4 GPU with 40 multiprocessors and 15,109 MB of total memory. 100 epochs were executed with a batch of 16 images. The training loss (MAE) was reduced from 0.1 in the first epoch to 0.03 in the last epoch, as can be seen in Figure 5.

48



49

50

51

Figure 5. Training set loss; source: Tensorboard

### 6.1.3 R-CNN Training

The whole implementation was done using the artificial intelligence lab at the Czech University of Life Sciences in Prague, because R-CNN has high memory requirements and caused Google Colaboratory to crash (due to lack of memory). The environment settings as seen in Table 5 were used:

IDE	VS Code with Jupyter Extension
Kernel	Python 3.8.12 within Anaconda
AI Framework	Tensorflow 2.7.0 for GPU
Nvidia Configuration	NVIDIA Quadro P400, cuda 11.2, cudnn 8.1

Table 5. R-CNN environment settings.

We have implemented region proposals with selective search using IoU (Intersection over Union) configured as seen in Table 6:

Max Samples	55 (based on the maximum in training set)
Selective Search Iterate Results	2000 (proposed in original paper)
IoU Object Limit	0.7
IoU Background Limit	0.3

Table 6. R-CNN configuration.

The images used were 224 x 224 in size. We chose a VGG model pre-trained on the ImageNet dataset (input layer, thirteen convolutional layers, five MaxPooling layers, Flatten, Dense). After encoding the label set once and splitting it into training (90%) and test sets (10%), we proceeded to train with the configurations as seen in Table 7. Early stopping caused the training process to stop after thirty-nine epochs.

Error Function	Binary cross-entropy
Optimizer	Adam
Learning Rate	0.0001
Training Epochs	100
Steps in Epoch	10
Patience Epochs for Early Stopping	20

Table 7. R-CNN training hyperparameters.

## 6.2 Utilities for Further Research

In order to ease the process of data creation for the next steps of the project, we developed three image and label processing tools: an image splitter, a vector visualization, and an image merger. These tools are available in the GitHub repository of our project.<sup>[8]</sup> The neural networks that we used for object detection accept square input, and if it is not square, the image is reshaped to a standard input size. For large tablets, there would be a significant loss of data, so it is necessary to slice large images into smaller, uniformly sized square images and train the network on these slices. We chose a fixed image size of 416 x 416 (a multiple of eight, which is generally better for machine learning purposes [Chollet and Pecinovsky 2019]).

While for the research presented in this article, we split the images before labelling, this slowed down the labelling process. Therefore, we developed an image splitter and an image merger. Our proposed system works as follows: after labelling, a large image with a cuneiform-bearing object is cut into squares with 50% overlap, so there is no data loss if strokes are on the edge of one square, since they are in the middle of the next one. Then the neural network predicts where the horizontal strokes are in the image. The networks return bounding boxes which indicate the location of the strokes. These bounding boxes are replaced by vectors of strokes in an empty image, and the strokes in the whole tablet are reconstructed using merging. In this way we can create an automatic vectorization of horizontal (and other) strokes in the whole tablet (see Figure 6).

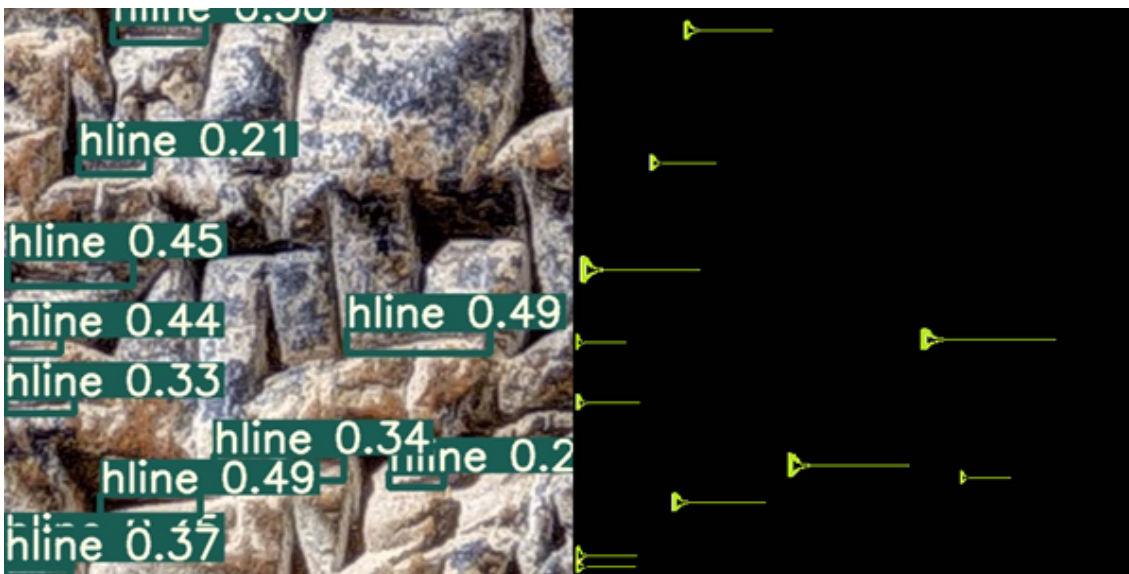


Figure 6. Output image from the vector visualisation tool, tablet YPM BC 021234.

The main challenge in preparing the set of tools was dealing with splitting labels. When splitting the image into smaller squares, there is a cut-off threshold for deciding whether the annotated strokes are still significant enough to be used in training. The threshold is based on a percentage that determines what portion of the annotated strokes can be kept and what should be removed.

## Acknowledgements

This research was funded by two grants. The project cuneiform analysis using Convolutional Neural Networks reg. no. 31/2021 was financed from the OP RDE project Improvement in Quality of the Internal Grant Scheme at CZU, reg. no. CZ.02.2.69/0.0/0.0/19\_073/0016944. The project no. RA200000010 was financed by the CULS – Ariel University cooperation grant.

## Notes

[1] See Hamplova, A., Franc, D., Pavlicek, J., Romach, A., Gordin, S., Cejka, M., and Vesely, A. (2022) "adelajelinkova/cuneiform", *GitHub*, available at: <https://github.com/adelajelinkova/cuneiform>.

- [2] See Ultralytics/Yolov5 (2021) "Yolov5", *GitHub*, available at: <https://github.com/ultralytics/yolov5>.
- [3] See Jocher, G., Chaurasia, A., and Qiu, J. (2023) "YOLO by Ultralytics (Version 8.0.0)", *GitHub*, available at: <https://github.com/ultralytics/ultralytics>.
- [4] See Bi, A. (2020) alankbi/detecto "Build fully-functioning computer vision models with PyTorch", *GitHub*, available at: <https://github.com/alankbi/detecto>.
- [5] See Approboflowcom (2021) "Roboflow Dashboard", *GitHub*, available at: <https://app.roboflow.com/>.
- [6] See tzutalin/labelIm (2021) "LabelIm", *GitHub*, available at: <https://github.com/tzutalin/labelIm>.
- [7] See Roboflow (2021) "Augmented Horizontal Dataset", available at: <https://app.roboflow.com/ds/t1bzmjvYH?key=qSadLELYkV>.
- [8] See <https://github.com/adelajelinkova/cuneiform/tree/main/Utilities>.

## Works Cited

- Anderson et al. 2023** Anderson, A. et al. (eds) (2023) *Proceedings of the ancient language processing workshop (RANLP-ALP 2023)*. Shoumen, Bulgaria: INCOMA Ltd.
- André-Salvini and Lombard 1997** André-Salvini, B. and Lombard, P. (1997) "La découverte épigraphique de 1995 à Qal'at al-Bahrein: un jalon pour la chronologie de la phase Dilmoun Moyen dans le Golfe arabe", *Proceedings of the seminar for Arabian studies, 1995*. pp. 165–170.
- Bertrand 2010** Bertrand, P. (2010) "Du De re diplomatica au Nouveau traité de diplomatique: réception des textes fondamentaux d'une discipline", in Leclant, J., Vauchez, A., and Hurel, D.O. (eds) *Dom Jean Mabillon, figure majeure de l'Europe des lettres: Actes des deux colloques du tricentenaire de la mort de Dom Mabillon (abbaye de Solesmes, 18-19 mai 2007)*, pp. 605-619. Paris: Académie des Inscriptions et Belles-Lettres.
- Bogacz and Mara 2018** Bogacz, B. and Mara, H. (2018) "Feature descriptors for spotting 3D characters on triangular meshes", *Proceedings of the 16th international conference on frontiers in handwriting recognition, IEEE, 2018*. Niagara Falls, NY, USA, 5-8 August. pp. 363-368. Available at: <https://ieeexplore.ieee.org/document/8583788>
- Bogacz and Mara 2022** Bogacz, B. and Mara, H. (2022) "Digital assyriology: Advances in visual cuneiform analysis", *Journal on Computing and Cultural Heritage*, 15(2). <https://doi.org/10.1145/3491239>.
- Bogacz, Gertz, and Mara 2015a** Bogacz, B., Gertz, M., and Mara, H. (2015a) "Character retrieval of vectorized cuneiform script", *Proceedings of the 13th international conference on document analysis and recognition, IEEE, 2015*. Piscataway, NJ, 23-26 August. pp. 326-330. <https://doi.org/10.1109/ICDAR.2015.7333777>
- Bogacz, Gertz, and Mara 2015b** Bogacz, B., Gertz, M., and Mara, H. (2015b) "Cuneiform character similarity using graphic representations", in Wohlfahrt, P. and Lepetit, V. (eds) *20th computer vision winter workshop*. Graz, Austria: Verlag der Technischen Universität Graz, pp. 105–112.
- Bogacz, Howe, and Mara 2016** Bogacz, B., Howe, N., and Mara, H. (2016) "Segmentation free spotting of cuneiform using part structured models", *Proceedings of the 15th international conference on frontiers in handwriting recognition, IEEE, 2016*. Shenzhen, China, 23-26 October. pp. 301-306. <https://doi.org/10.1109/ICFHR.2016.0064>.
- Bramanti 2015** Bramanti, A. (2015) "The cuneiform stylus. Some addenda", *Cuneiform digital library notes*, 2015(12). Available at: <https://cdli.mpiwg-berlin.mpg.de/articles/cdln/2015-12> (Accessed: 26 January 2024).
- Brusius 2015** Brusius, M. (2015) *Fotografie und Museales Wissen: William Henry Fox Talbot, das Altertum und die Absenz der Fotografie*. Berlin: De Gruyter.
- Cammarosano 2014** Cammarosano, M. (2014) "The cuneiform stylus", *Mesopotamia: Rivista di Archeologia, Epigrafia e Storia Orientale Antica*, 69, pp. 53–90.
- Cammarosano et al. 2014** Cammarosano, M. et al. (2014) "Schriftmetrologie des Keils: Dreidimensionale Analyse von Keileindrücken und Handschriften", *Die Welt des Orients*, 44(1), pp. 2-36.
- Cho et al. 2016** Cho, J. et al. (2016) "How much data is needed to train a medical image deep learning system to achieve necessary high accuracy?" *arXiv*, 1511.06348. Available at: <http://arxiv.org/abs/1511.06348>.
- Chollet and Pecinovský 2019** Chollet, F. and Pecinovský, R. (2019) *Deep learning v Jazyku Python: Knihovny Keras, TensorFlow*. Praha, Prague: Grada Publishing a.s.
- Cohen et al. 2004** Cohen, J. et al. (2004) "iClay: Digitizing cuneiform", *Proceedings of the 5th international symposium on virtual reality, archaeology and cultural heritage, EG, 2004*. Oudenaarde, Belgium, 7-10 December. pp. 135-143. Available at: <https://doi.org/10.2312/VAST/VAST04/135-143>.
- Collins et al. 2019** Collins, T. et al. (2019) "Automated low-cost photogrammetric acquisition of 3D models from small form-factor artefacts", *Electronics*, 8, p. 1441. <https://doi.org/10.3390/electronics8121441>.
- Dahl, Hameeuw, and Wagensohn 2019** Dahl, J.L., Hameeuw, H., and Wagensohn, K. (2019) "Looking both forward and back: imaging cuneiform", *Cuneiform digital library preprints* [Preprint]. Available at: <https://cdli.mpiwg-berlin.mpg.de/articles/cdli/14.0>.
- Dencker et al. 2020** Dencker, T. et al. (2020) "Deep learning of cuneiform sign detection with weak supervision using transliteration alignment", *PLOS ONE*, 15(12), p. e0243039. <https://doi.org/10.1371/journal.pone.0243039>.
- Duranti 1998** Duranti, L. (1998) *New uses for an old science*. Chicago, IL: Scarecrow Press.
- Earl et al. 2011** Earl, G. et al. (2011) "Reflectance transformation imaging systems for ancient documentary artefacts", in Dunnand, S., Bowen, J.P., and Ng K.C. (eds) *EVA London 2011: Electronic visualisation and the arts*, pp. 147-154. London: BCS.
- Edzard 1980** Edzard, D.O. (1980) "Keilschrift", *Reallexikon der Assyriologie*, 5, pp. 544-568.

- Fisseler et al. 2013** Fisseler, D. et al. (2013) "Towards an interactive and automated script feature Analysis of 3D scanned cuneiform tablets", *Proceedings of the scientific computing and cultural heritage conference, 2013*. Heidelberg, Germany, 18-20 November. Available at: [https://www.researchgate.net/publication/267921266\\_Towards\\_an\\_interactive\\_and\\_automated\\_script\\_feature\\_Analysis\\_of\\_3D\\_scanned\\_cuneiform\\_tablets](https://www.researchgate.net/publication/267921266_Towards_an_interactive_and_automated_script_feature_Analysis_of_3D_scanned_cuneiform_tablets).
- Fisseler et al. 2014** Fisseler, D. et al. (2014) "Extending philological research with methods of 3D computer graphics applied to analysis of cultural heritage", *Proceedings of the eurographics workshop on graphics and cultural heritage, 2014*. Darmstadt, Germany, 6-8 October, pp. 165-172. <https://doi.org/10.2312/gch.20141314>.
- Girshick 2015** Girshick, R. (2015) "Fast R-CNN", *Proceedings of the IEEE international conference on computer vision, IEEE, 2015*. Santiago, Chile, 11-18 December. pp. 1440-1448. <https://doi.org/10.1109/ICCV.2015.169>.
- Girshick et al. 2014** Girshick, R. et al. (2014) "Rich feature hierarchies for accurate object detection and semantic segmentation", *Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE, 2014*. Columbus, OH, USA, 23-28 June. <https://doi.org/10.1109/CVPR.2014.81>.
- Gordin and Romach 2022** Gordin, S. and Romach, A. (2022) "Optical character recognition for complex scripts: A case-study in cuneiform", *Proceedings of the alliance of digital humanities organizations conference, IDHC, 2022*. Tokyo, Japan, 25-29 July. Available at: <https://dh-abstracts.library.cmu.edu/works/11708>.
- Gordin et al. 2020** Gordin, S. et al. (2020) "Reading Akkadian cuneiform using natural language processing", *PLOS ONE*, 15(10). <https://doi.org/10.1371/journal.pone.0240511>.
- Gutherz et al. 2023** Gutherz, G. et al. (2023) "Translating Akkadian to English with neural machine translation", *PNAS Nexus*, 2(5). <https://doi.org/10.1093/pnasnexus/pgad096>.
- Hameeuw and Willems 2011** Hameeuw, H. and Willems, G. (2011) "New visualization techniques for cuneiform texts and sealings", *Akkadica*, 132(3), pp. 163-178.
- Jiao et al. 2019** Jiao, L. et al. (2019) "A survey of deep learning-based object detection", *IEEE Access*, 7, pp. 128837-128868. <https://doi.org/10.1109/ACCESS.2019.2939201>.
- Jocher et al. 2020** Jocher, G. et al. (2020) "ultralytics/yolov5: Initial release", *Zenodo*. <https://doi.org/10.5281/zenodo.3908560>.
- Kriege et al. 2018** Kriege, N.M. et al. (2018) "Recognizing cuneiform signs using graph based methods", *Proceedings of the international workshop on cost-sensitive learning, PMLR, 2018*. San Diego, CA, USA, 5 May. pp. 31-44. Available at: <https://proceedings.mlr.press/v88/kriege18a.html>.
- Labat 1988** Labat, R. and Malbran-Labat, F. (1988) *Manuel d'épigraphie akkadienne*, 6th ed. Paris: Librairie Orientaliste Paul Geuthner.
- Mara and Krömker 2013** Mara, H. and Krömker, S. (2013) "Vectorization of 3D-characters by integral invariant filtering of high-resolution triangular meshes", *Proceedings of the international conference on document analysis and recognition, IEEE, 2013*. Washington, DC, USA, 25-28 August. pp. 62-66. <https://doi.org/10.1109/ICDAR.2013.21>.
- Mara et al. 2010** Mara, H. et al. "GigaMesh and Gilgamesh: 3D multiscale integral invariant cuneiform character extraction", *Proceedings of the 11th international symposium on virtual reality, archaeology, and cultural heritage, EG, 2010*. Paris, France, 21-24 September. <https://doi.org/10.2312/VAST/VAST10/131-138>.
- Massa et al. 2016** Massa, J. et al. (2016) "Cuneiform detection in vectorized raster images", *Proceedings of the 21st computer vision winter workshop, 2016*. Rimske Toplice, Slovenia, 3-5 February. Available at: <https://d-nb.info/1191851524/34>.
- Mishra 2022** Mishra, D. (2022) "Deep learning based object detection methods: A review", *Medicon Engineering Themes*, 2(4). <https://doi.org/10.55162/MCET.02.027>.
- Pavliček 2018** Pavliček, J. et al. (2018) "Automated wildlife recognition", *AGRIS on-line papers in economics and informatics*, 10(1), pp. 51-60. <https://doi.org/10.7160/aol.2018.100105>.
- Punia et al. 2020** Punia, R. et al. (2020) "Towards the first machine translation system for Sumerian transliterations", *Proceedings of the 28th international conference on computational linguistics, ACL, 2020*. Barcelona, Spain, 13-18 September. pp. 3454-3460. <https://doi.org/10.18653/v1/2020.coling-main.308>.
- Redmon et al. 2016** Redmon, J. et al. (2016) "You only look once: Unified real-time object detection", *Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE, 2016*. Las Vegas, NV, USA, 26 June-1 July. <https://doi.org/10.1109/CVPR.2016.91>.
- Ren et al. 2015** Ren, S. et al. (2015) "Faster R-CNN: Towards real-time object detection with region proposal networks", *Proceedings of the advances in neural processing systems conference, NeurIPS, 2015*. Montreal, Canada, 7-12 December. Available at: [https://papers.nips.cc/paper\\_files/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html](https://papers.nips.cc/paper_files/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html).
- Rothacker et al. 2015** Rothacker, L. et al. (2015) "Retrieving cuneiform structures in a segmentation-free word spotting framework", *Proceedings of the 3rd international workshop on historical document imaging and processing, ACM, 2015*. Nancy, France, 22 August. pp. 129-136. <https://doi.org/10.1145/2809544.2809562>.
- Rusakov et al. 2019** Rusakov, E. et al. (2019) "Generating cuneiform signs with cycle-consistent adversarial networks", *Proceedings of the 5th international workshop on historical document imaging and processing, ACM, 2019*. Sydney, Australia, 20-21 September. pp. 19-24. <https://doi.org/10.1145/3352631.3352632>.
- Rusakov et al. 2020** Rusakov, E. et al. (2020) "Towards query-by-expression retrieval of cuneiform signs", *Proceedings of the 17th international conference on frontiers in handwriting recognition, IEEE, 2020*. Dortmund, Germany, 7-10 September. pp. 43-48. <https://doi.org/10.1109/ICFHR2020.2020.00019>.
- Sommerschield et al. 2023** Sommerschield, T. et al. (2023) "Machine learning for ancient languages: A survey", *Computational Linguistics*, 49(3), pp. 1-44. [https://doi.org/10.1162/coli\\_a\\_00481](https://doi.org/10.1162/coli_a_00481).
- Streck 2010** Streck, M.P. (2010) "Großes Fach Altorientalistik: Der Umfang des keilschriftlichen Textkorpus", *Mitteilungen der Deutschen Orient-Gesellschaft*, 142, pp. 35-58.

**Taylor 2015** Taylor, J. (2014) "Wedge order in cuneiform: A preliminary survey", in Devecchi, E., Müller, G.G.W., and Mynářová, J. (eds) *Current research in cuneiform palaeography: Proceedings of the workshop organised at the 60th rencontre assyriologique internationale, Warsaw, Poland, 2014*, pp. 1–30. Gladbeck, Germany: PeWe-Verlag.

**Wagensonner 2015** Wagensonner, K. (2015) "On an alternative way of capturing RTI images with the camera dome", *CDLN*, 2015(1), pp. 1-12.

**Wevers and Smits 2019** Wevers, M. and Smits, T. (2020) "The visual digital turn: Using neural networks to study historical images", *Digital Scholarship in the Humanities*, 35(1), pp. 194-207. <https://doi.org/10.1093/lc/fqy085>.

**Yamauchi et al. 2018** Yamauchi, K., Yamamoto, H., and Mori, W. (2018) "Building a handwritten cuneiform character image set", *Proceedings of the 11th international conference on language resources and evaluation, ACL, 2018*. Miyazaki, Japan, 7-12 May. pp. 719-722. Available at: <https://aclanthology.org/L18-1115>.



This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.