Exploring Learning Approaches for Ancient Greek Character Recognition with Citizen Science Data

Matthew I. Swindall[1], Gregory Croisdale[2], Chase C. Hunter[2], Ben Keener[3], Alex C. Williams[2],
James H. Brusuelas[3], Nita Krevans[4], Melissa Sellew[4], Lucy Fortson[4], John F. Wallin[1]
Middle Tennessee State University[1], University of Tennessee[2],
University of Kentucky[3], University of Minnesota[4]
{mis2n@mtmail.mtsu.edu,gcroisda@vols.utk.edu,thunte11@vols.utk.edu,bdke225@uky.edu,acw@utk.edu
james.brusuelas@uky.edu,nkrevans@umn.edu,sellew@umn.edu,lfortson@umn.edu,john.wallin@mtsu.edu}

Abstract—The central dogma of handwritten character recognition remains inextricably linked to optical character recognition methods for print media. Alongside their reliance on proprietary data and lack of open-access software, the applicability of these optical character recognition methods to handwritten characters from low-quality documents (e.g., that are damaged) remains unknown. In this paper, we compare and contrast the performance of state-of-the-art optical character recognition tools for print and learning models engineered with state-of-the-art machine learning toolkits trained on handwritten inputs. Using Tesseract OCR as a baseline, we build, optimize, and evaluate three types of convolutional neural networks that are trained on the `AL-ALL` and `AL-PUB` datasets, a collection of images of handwritten ancient Greek characters that were labeled by volunteers through the *Ancient Lives* online citizen science project. We find our best-performing machine learning model to be 92.57% accurate compared to Tesseract OCR's 11.15%. Following our analysis, we present a brief examination of our models' shortcomings, introduce the publicly-available `AL-PUB` dataset, and, describe *Theia*, a web-based tool that democratizes our machine learning models for public use. We conclude by discussing the promise of our findings for advancing research at the intersection of machine learning, manuscript transcription, and the digital humanities.

Keywords—Ancient Greek; character transcription; machine learning; papyrology; crowdsourcing; citizen science; dataset.

I. Introduction

LABELED datasets of handwritten digits and characters, such as MNIST, have been critical in advancing the field of machine learning over the past three decades [1], [2], [3]. Within the past few years, studies have continued to acknowledge the relevance of such datasets with particular interests in extending them (e.g. from digits to letters [4]). When data is unavailable, character recognition researchers are generally required to create their own datasets [5], which can often be costly in terms of time, effort, and money. Today, the number of openly available datasets remains significantly limited in quantity despite their growing demand, particularly in cultural heritage contexts [6], [7].

Over the past decade, one cost-effective and increasingly common method of data collection is crowdsourcing, in which a gold standard label is generated by multiple annotators [8]. Despite being generally cost-effective, the use of multiple annotators often results in a set of noisy labels that are non-uniform and maintain some level of disagreement. Such noise has been shown to heavily influence a dataset's utility in machine learning contexts [9], [10], [11], and research has therefore given significant attention to engineering techniques to mitigate noise by various statistical measures [12], [13], [14]. Other approaches have simply thrown out the noisy labels altogether [15]. Modern datasets (e.g. MNIST) generally fail to reflect the reality that crowdsourced data is not only imperfect, but so large in magnitude that identifying annotator errors is both challenging and time-consuming in task settings in which a ground truth label may be ambiguous. One such example is transcribing the deteriorated papyrus manuscript in Figure 1.

Fig. 1: An example from the Oxyrhynchus papyri collection.

In this paper, we compare and contrast the effectiveness of a state-of-the-art optical character recognition tool (i.e., Tesseract) to a set of novel machine learning approaches that share the task of classifying handwritten character images. Our approaches are fueled by the *Ancient Lives* dataset, a collection of digitized images of handwritten ancient Greek characters that are the product of the *Ancient Lives* crowdsourcing initiative in which volunteers annotated digital images of ancient papyrus manuscripts. We first present a novel cropping algorithm that isolates and extracts each character in each manuscript into independent and labeled image files. Using the *Ancient Lives* dataset as source material, we establish two ancient Greek character image datasets: (1) `AL-ALL` and (2) `AL-PUB`. We use these two datasets to train three unique machine learning models and compare their effectiveness against the Tesseract OCR tool. We find that all three model approaches perform more effectively than Tesseract, with our best-performing model achieving an accuracy of 92.73%. Following an analysis of our results, we conduct

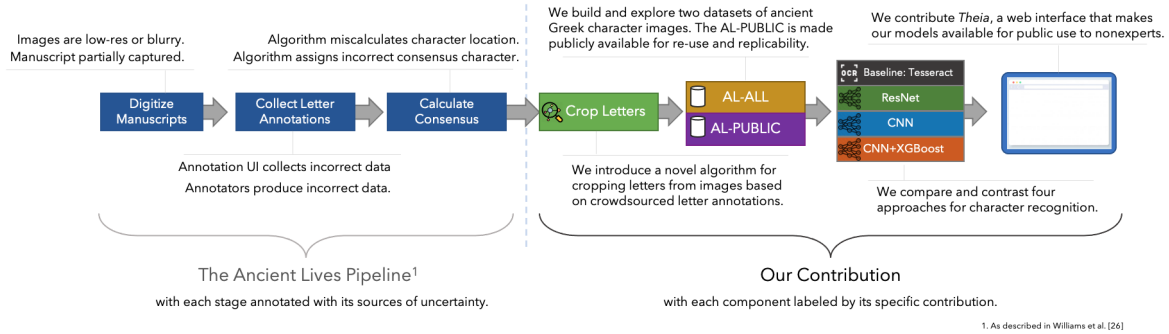


Fig. 2: A visual representation of our contributions and how they build on prior work.

an audit of misclassifications across our model and introduce *Theia*, a web interface that allows nonexperts (e.g., in the humanities) to utilize our classification models without the need for expertise. We conclude with a discussion on our findings and their implications for research at the intersection of machine learning, citizen science, noisy labeling, and handwritten character recognition in cultural heritage contexts.

II. Related Literature

A. Handwritten Character Recognition

For more than three decades, advances in handwritten character recognition have relied substantially on publicly available datasets. Amid the range of datasets, the most common character recognition dataset to date is MNIST[1], a dataset of 70,000 images of handwritten digits written by high school students and employees of the United State Census Bureau. The dataset has served as a focal point of application, education, and extension in machine learning [4]. Following the widespread usage of MNIST, a myriad of character recognition datasets have been collected and made publicly available, such as those that focus on handwriting in multiple languages [16], [17], [18], symbols [19], [20], or noisy handwriting [21].

B. Crowdsourcing and Noise Labelers

Crowdsourcing is an exceptionally popular technique for building datasets both in and beyond computer vision. As a by-product of engaging labelers of varying backgrounds, expertise, and personal characteristics, crowdsourcing research is often concerned with techniques for managing noisy labels [9], [10], [11]. Controlled studies have, for example, focused substantially on the development of characterizing the reliability of annotators, particularly in cultural heritage projects where finances (i.e., for labeling) are often very limited [14]. To mitigate the concerns around cost and reliability, humanities-oriented transcription projects, such as Old Weather [22] and Operation War Diary [23], generally rely on volunteers who are intrinsically motivated to participate in labeling as opposed to other types of labelers who are extrinsically motivated (e.g. Amazon Mechanical Turk workers) [24]. Despite the sustained use of crowdsourcing for data collection in manuscript contexts, handwritten character recognition datasets remain relatively limited within the purview of ancient manuscripts, alongside the digital humanities at large.

[1]https://en.wikipedia.org/wiki/MNIST_database

C. Crowdsourcing Manuscript Transcription

Enlisting the help of the public, either through monetary or voluntary means, has been a growing topic of interest for the digital humanities in recent years. Many such projects are geared toward the production of digital transcriptions that are of interest both to archival repositories and to scholars working on critical editions. Transcribe Bentham [25], for example, is a long-running and thriving project that enlists the help of the public to transcribe the journal pages of Jeremy Bentham. Beyond the task of document transcription, newer projects, such as Ancient Lives and Scribes of the Cairo Geniza, employ their crowdsourced transcriptions to identify or simply contextualize unstudied manuscripts [26], which may require specialized computational pipelines for data processing and preparation [27]. The popularity of this kind of research is evidenced by the rise of platforms dedicated to hosting projects for archives, libraries, and museums since 2010, most notably the *Zooniverse* citizen science platform [28], the *From the Page* platform [29], and the *SciFrabric* / Pybossa platform [30]. While crowdsourced manuscript transcription efforts and initiatives have risen in popularity, the public availability and re-use of crowdsourced transcription data from these projects has yet to become the norm.

D. Contribution

This paper makes several contributions, each of which expands on the prior literature in a unique way. First, we contribute a novel cropping algorithm that extracts individual character images from manuscript images using crowdsourced data. Second, we introduce two datasets of ancient Greek character images, one of which is made publicly available for re-use and replicability among the digital humanities, computer vision, and machine learning communities. Third, we design, implement, and optimize three machine learning modelling approaches that are trained with state-of-the-art machine learning toolkits. Alongside these models, we contribute an evaluation of these models, focusing specifically on their ability to classify ancient Greek characters in comparison to a state-of-the-art OCR tool. Our final contribution is *Theia*, a web interface tool that allows nonexperts to utilize our machine learning models with their own character images. An overview of our contributions is shown in Figure 2.

Fig. 3: The Ancient Lives interface.

III. Ancient Lives

Ancient Lives was a web-based citizen science project that was launched in June 2011 in coordination with the Zooniverse [28] and concluded in June 2018. Through the internet, the project enlisted the help of volunteers from across the world to transcribe deteriorated ancient Greek papyrus fragments (*i.e.*, remnants of a larger manuscript). All 12,070 papyri fragments that were transcribed via the Ancient Lives system belonged to the family of the Oxyrhynchus papyri, an established collection of ancient Greek papyrus manuscripts discovered in the ancient Egyptian city of Oxyrhynchus [31], [32], [33].

A. Task Interface

Counter to most citizen science projects for transcription, Ancient Lives's task interface treats a transcription event as an object detection process. Users are asked to transcribe each fragment by finding one letter at a time. At the beginning of a task, users are presented with a papyrus fragment along with a virtual keyboard that allows the user to customize their transcription experience (*e.g.*, change annotation color). Annotations can be added to the interface simply by clicking on the image. An annotation's position can be updated at any time by clicking and dragging the UI element to a new location. Once an annotation is created, a user can assign a letter to the annotation, or update the annotation's letter, by clicking on the appropriate letter on the virtual keyboard. When a user hovered over any key on the virtual keyboard, two example images of the Greek letter or symbol were shown on the top-left panel of the keyboard. Lastly, users are given a mini-map to show the field-of-view of the papyrus fragment being viewed. Users can change their field-of-view of the image by clicking and dragging on either the mini-map or the image itself. The interface enforced no constraints about how (e.g. in what order) letters or symbols should be annotated. The interface is shown in Figure 3.

Fig. 4: Examples of each character in the dataset.

B. Processing Pipeline & Data Quality

To organize and collate the wealth of annotation data, an existing computational pipeline developed by Williams et al. [27] was leveraged. The pipeline implements several algorithms that facilitate the processes of (1) aggregating letter annotations into consensus annotations and (2) creating "chains" of letter annotations to create strings of text.

In the creation of the dataset at hand, the team only makes use of (1) as the research is limited to individual characters. The quality of the data produced through the pipeline's procedure has been vetted [27], and the Ancient Lives data itself has been used toward several other contexts (e.g., deteriorated manuscript identification [26]).

Character	Count	Character	Count
Alpha (A, α)	42,538	**Nu** (N, ν)	44,896
Beta (B, β)	2,534	**Xi** (Ξ, ξ)	1,201
Gamma (Γ, γ)	6,907	**Omicron** (O, o)	46,334
Delta (Δ, δ)	11,716	**Pi** (Π, π)	17,112
Epsilon (E, ϵ)	31,581	**Rho** (P, ρ)	20,448
Zeta (Z, ζ)	1,425	**Sigma** (Σ, σ)	62
Eta (H, η)	15,062	**Tau** (T, τ)	32,034
Theta (Θ, θ)	7,575	**Upsilon** (Y, υ)	15,762
Iota (I, ι)	25,593	**Phi** (Φ, ϕ)	6,063
Kappa (K, κ)	17,932	**Chi** (X, χ)	9,155
Lambda (Λ, λ)	13,253	**Psi** (Ψ, ψ)	904
Mu (M, μ)	13,225	**Omega** (Ω, ω)	16,043

TABLE I: Counts for each letter in the Ancient Lives dataset.

C. Cropping Algorithm and Consensus Label

The *Ancient Lives* web interface was used to collect annotations on manuscript images that included one or more characters. To generate a dataset of characters, we designed a cropping algorithm that was applied to each manuscript image to extract individual character images. Using the the coordinate information of each annotation, the distance to the nearest adjacent character (δ) is calculated. Each image was then cropped by $1.1(\frac{\delta}{2})$ pixels including a 10% buffer, from the indicated location of the character along the vertical and horizontal dimensions. All cropped images were automatically resized to 70 x 70 pixels. To limit the number of extraction errors (e.g. extracting markings), a filtering criterion was applied in which images that had fewer than three annotators in agreement were removed. As the focus is on alphabetic characters, symbols and miscellaneous markings were excluded. Target labels for each image were chosen by taking the consensus of annotators' labels (i.e., majority vote). Table I shows the dataset's character distribution.

D. The Ancient Lives Dataset

We used the cropping algorithm to produce two versions of the *Ancient Lives* dataset for training machine learning models:

- `AL-PUB`: Includes 195,683 labeled character images from 5,043 published manuscript images that were used used in the Ancient Lives web interface.
- `AL-ALL`: Includes 399,330 labeled character images from 12,070 published and unpublished manuscript images that were used in the Ancient Lives web interface. This includes all images from the `AL-PUB` dataset.

Our motivation to create two versions of the *Ancient Lives* dataset was driven by one issue: scholars working on unpublished and unidentified manuscripts. In order not to reveal manuscript data still under papyrological research, the AL-PUB dataset stems from previously published material in The Oxyrhynchus Papyri Series. Nevertheless, each image in both dataset versions *ideally* contains one tightly cropped Greek character, stored in JPEG format, and can be of a range of several resolutions. All 24 Greek alphabet characters are represented in the dataset. Images are sorted into sub-directories for each character and follows a file-naming convention that enumerates its associated labels.

IV. Methods

The goal of our research is to compare and contrast the performance of learning approaches trained on the Ancient Lives dataset to state-of-the-art optical character recognition methods. Here, we describe the various modelling approaches we explored alongside the OCR tool used as a baseline measure.

A. Baseline: Tesseract

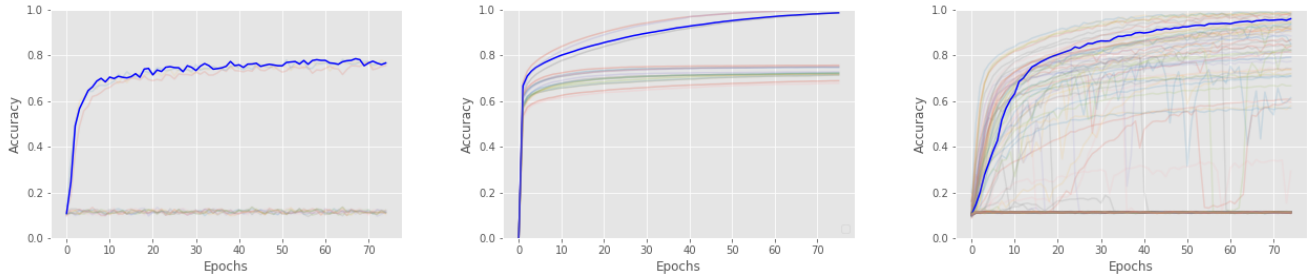
Tesseract [34] is a state-of-the-art optical recognition software tool for extracting text from images. Prior examinations of Tesseract have demonstrated its effectiveness for character recognition for extracting text across a multitude of languages [35], including ancient Greek texts [36], [37]. Further, comparative studies suggest that the effectiveness of Tesseract varies between contexts (e.g., license plates in greyscale vs. in color) in contrast to proprietary OCR alternatives (e.g. Transym) [38]. Based on the wealth of prior research reinforcing its utility, we used the Tesseract engine designed for ancient Greek[2] as our baseline measure. We ran Tesseract on individual character page segmentation mode with a whitelist consisting solely of the characters in the dataset. Despite being on single character mode, Tesseract may return multiple characters. To remedy this without penalizing Tesseract, we considered any transcription with the target character present to be accurate.

B. Learning Approaches

We designed and implemented three unique learning approaches using Tensorflow and Keras that stem from the broader family of Convolutional Neural Networks (CNNs). The decision to employ CNNs as an alternative method to optical character recognition tools was motivated by the wealth of literature that has demonstrated the success of CNNs as tools for unique handwritten character recognition scenarios [39], [40], [41]. Our explored learning approaches include:

1) Standard CNN (`CNN-BASE`): A standard CNN with a configurable architecture of convolution layers (e.g., that control image tensor dimensionality) and max-pooling layers. This base architecture is both highly configurable and highly performant for character recognition [39].
2) CNN + XGBoost (`CNN-XGB`): XGBoost is a machine learning library which applies gradient boosted trees in classifying data. When layered with a standard CNN, XGBoost is able to classify characters with more depth and greater accuracy using the extracted features from the CNN. The concept is modified from ConvXGB [42], with some modifications to hyperparameters tuned.
3) ResNet Model (`RESNET`): We based this model off of the Residual Learning framework first introduced in 2015 [43]. This framework is a modification of the typical CNN used in image recognition, where a residual learning component is added to combat problems when increasing depth. This in turn allows the model to remain time efficient and accurate at above average depth, resulting in more detail captured from our character.

[2]https://ancientgreekocr.org/



(a) CNN−BASE (b) CNN−XGB (c) RESNET

Fig. 5: Visualizations of the grid search for the CNN−BASE, CNN−XGB, and RESNET models over 75 epochs using AL−TUNE. In contrast to the other models, CNN−XGB produces a smoother curve due to its decision tree design. Best models in blue.

C. Hyperparameter Tuning: Procedure

We optimized our machine learning models by conducting both a coarse grid search and fine grid search on the standard list of configurable hyperparameters for convolutional neural network models [44]. Our initial plan of execution involved the exploration of multiple values for all hyperparameters across each model. However, preliminary runs of the hypertuning process using the AL−PUB and the AL−ALL datasets were estimated to require a compute time of upward of a year to execute to completion. To reduce the amount of time required to conduct an adequate grid search, we conducted a hypertuning process that utilized a reduced dataset of character images alongside a smaller number of explored parameters.

1) AL-TUNE: A Reduced Dataset of Character Images: We randomly sampled 39,924 (10%) character images from the AL−ALL dataset to create a derivative dataset, which we henceforth refer to as AL−TUNE. After the sampling process had completed, we confirmed that character distributions between the AL−ALL and AL−TUNE datasets were relatively similar, suggesting that the reduced dataset was statistically representative of the larger dataset. While representation was generally maintained, several characters (i.e., Sigma, Psi) were not well-represented in the reduced AL−TUNE dataset as a by-product of having significantly lower representation in the larger AL−ALL dataset.

2) Reduction of Explored Values: A preliminary grid search using the AL−TUNE dataset with all three model types revealed that several hyperparameters (i.e., activation function, convolutional kernel size, max pooling kernel size, and momentum) had minimal effect on the outcome of validation accuracy across searches. We therefore reduced the range of explored values for these hyperparameters by setting them to static values as shown in Table III. The range of explored values for other hyperparameters was guided by tool documentation and prior studies that suggest appropriate values for configuring CNNs [44]. The complete list of explored hyperparameter values, alongside best-performing value for each hyperparameter for each model, are shown in Table III. The procedure was executed on a five-node GPU cluster with Intel i9 9820x processors and dual NVIDIA RTX 2080TI GPUs. All runs were conducted on individual cluster nodes, and each model took approximately 48 hours to complete.

D. Hyperparameter Tuning: Results

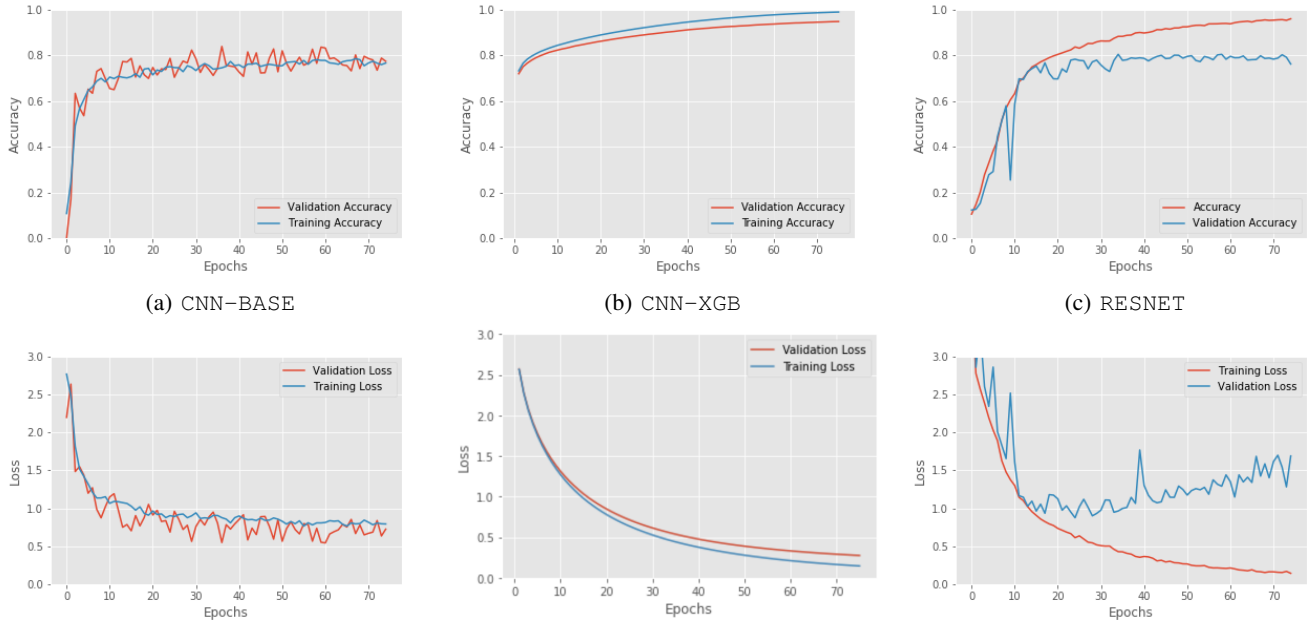
Figure 5 shows the validation accuracy for all hyperparameter configurations for each model across 75 training epochs. Throughout the grid search procedure, we observed significant variance in accuracy across the various model combinations. We specifically observed that, when the learning rate of a CNN was greater than 0.001, model accuracy tended to stagnate around 10%. Additionally, the CNN−XGB model's accuracy across 75 epochs tended to be positively correlated with the maximum tree depth. We chose the best models from this hyperparameter tuning procedure by averaging the 90th percentile of the accuracies that were generated by each epoch, which are shown in Table II. Following the conclusion of the hyperparameter tuning procedure, the best-performing model configurations were statically implemented for each model and subsequently executed on the same cluster system using both the AL−ALL and AL−PUB datasets.

TABLE II: Accuracy for each model during hypertuning.

Model	Val. Accuracy
CNN−BASE	86.7%
CNN−XGB	84.4%
RESNET	80.4%.

	Hyperparameter	Explored Values	Best
CNN−BASE	Learning Rate	0.001, 0.01, 0.03, 0.05	0.001
	Optimizer	Adam, RMSprop	Adam
	Number of Filters	16, 32, 64, 96	96
	Activation Func.	RELU	RELU
	Convolutional Kernel	3x3	3x3
	Max Pooling Kernel	2x2	2x2
CNN−XGB	Learning Rate (CNN)	0.001, 0.01, 0.1	0.001
	Number of Filters (CNN)	8, 16, 32, 64	16
	Eta (XGB Learning Rate)	0.01, 0.1, 0.2, 0.3, 0.4	0.1
	Max_Tree_Depth	5, 10, 30, 50, 100, 200	100
	Min_Child_Weight	2, 4, 6, 8, 10	10
RESNET	Learning Rate	0.001, 0.01, 0.03, 0.05	0.001
	Num. of Hidden Layers	2, 4, 8, 16, 32, 50	32
	Optimizer	Adam, RMSprop	RMSprop
	Number of Filters	16, 32, 64, 96	96
	Activation Function	RELU	RELU
	Kernel Size	3x3	3x3
	Momentum	0.9	0.9

TABLE III: An overview of the explored hyperparameter space alongside the best-performing values for each model.



(a) CNN−BASE (b) CNN−XGB (c) RESNET

(d) CNN−BASE (e) CNN−XGB (f) RESNET

Fig. 6: Accuracy and loss of the CNN−BASE, CNN−XGB, and RESNET models over 75 epochs on the AL−ALL dataset.

V. Results

In this section, we discuss the the results of all four explored approaches for both the AL−ALL dataset and the AL−PUB dataset. As Tesseract uses a pre-trained model, we discuss its accuracy both in aggregate and at the letter level. In contrast to Tesseract's discussion of results, we employed a standrad k-fold cross validation procedure with each of the three learning approaches. For each of these approaches, we discuss validation accuracy and loss and reported the observation accuracy. We conclude this section by revisiting instances of misclassifications to better understand both why and how our explored modeling approaches fail.

A. Tesseract

Overall, the pretrained Tesseract Ancient Greek model performed significantly worse than any of the newly trained model alternatives. Tesseract correctly produced a text transcription (i.e., that included the expected Greek character) in 45,988 (11.15%) of the character images in the AL−ALL dataset. Across this same dataset, Tesseract failed to recognize any text whatsoever in 133,950 character images (33.54%). Mirroring the tool's performance on AL−ALL dataset, Tesseract correctly produced a text transcription in 20,027 (10.23%) of the character images in the AL−PUB dataset while failing to recognize any text at all in 67,186 character images (34.33%). In examining the recognition effectiveness of individual letter images in the AL−ALL dataset, we observe that Tesseract was most effective at recognizing Epsilon images, classifying 4,576 of the 31,581 Epsilon images (14.49%) correctly. In contrast, we find that Beta images were the most frequently failure case for Tesseract, classifying only 68 of the 2534 Beta images (2.68%) correctly.

B. CNN-BASE, CNN-XGB, and RESNET

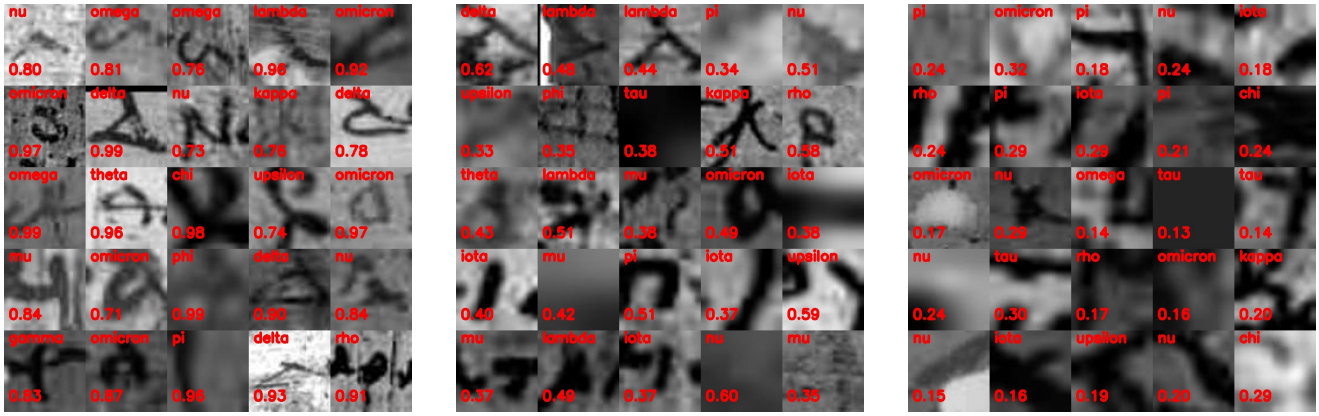
Table IV shows the average validation accuracy for each of the three model types for the standard k-fold cross-validation procedure over 10 iterations. Figure 6 shows accuracy and loss for model training across each of the three model types.

In general, all three modeling approaches significantly outperformed Tesseract OCR. Among the three learning models, the RESNET model achieved the best performing accuracy for both the AL−ALL and AL−PUB datasets, achieving an accuracy of 92.73% and 92.57% respectively. However, the RESNET model's accuracy yielded the highest deviation in performance among the three models. In contrast, the CNN−XGB exhibited significant variance between datasets, achieving an accuracy of 80.32% on the AL−PUB dataset and a substantially higher accuracy of 90.81% on the AL−ALL dataset. The CNN−BASE model averaged a similarly lower accuracy of 80.24% on the AL−PUB dataset while achieving a lower average of accuracy of 80.79% on the AL−ALL dataset.

The RESNET model's sustained accuracy across datasets suggests that it is the most reliable model for use in practice. However, all three approaches demonstrate a significant level of practical utility as the smallest average of accuracy among our models is 80.32%, which indicates – in the worst case – 38,510 of the AL−PUB's 195,683 character images were on average incorrectly classified.

TABLE IV: Average validation accuracy for all three models with both datasets across a 10-fold cross-validation procedure.

Model	AL−PUB	AL−ALL
CNN−BASE	82.24% (σ=0.03)	80.79% (σ=0.01)
CNN−XGB	80.32% (σ=0.63)	90.81% (σ=1.21)
RESNET	92.57% (σ=4.23)	92.73% (σ=3.44)



(a) 66% or Greater Confidence. (b) 33-66% Confidence. (c) 0-33% Confidence.

Fig. 7: Images that were misclassified by RESNET and CNN−BASE with confidence scores and consensus labels in red.

C. Auditing Misclassifications with RESNET and CNN-BASE

To better understand the shortcomings and failures of our learning approaches, we conducted a high-level examination of the errors encountered in our best-performing model: the RESNET model. In assessing the model's misclassifications, we observe that the RESNET model incorrectly classified a total of 29,533 character images (7.4%) out of the total 399,330 character images in the AL−ALL dataset. In comparison, the CNN−BASE model incorrectly classified 53,427 character images in the same dataset (13.1%). A total of 17,780 character images were misclassified by both of these models. Figure 7 shows a collage of examples across these models with confidence scores and consensus labels in red.

To better understand these misclassifications, we examined the characters that were misclassified by both the RESNET model and by the CNN-BASE model. We direct our investigation toward the 1,405 misclassified character images that were labeled as the Greek letter Alpha (α) according to our labeling pipeline shown in Figure 2. Restricting our sample to confidence levels greater than 33% (i.e., Figures 7a and 7b), we observe that misclassified character images can be categorized into two categories: (1) images that were misclassified by the volunteers or (2) images that are blank or unreadable. Based on a small sample of 100 characters in this low confidence group that were labeled as Alpha, but classified as another Greek character, 97 of the testing set images were found to be mislabeled. In other words, the RESNET model missed only three of the characters that were actually Alpha. In manually reviewing this small sample, we find the RESNET model to be more effective at classifying image examples of high ambiguity (e.g., cursive characters) than our research team members who lack formal training in Greek paleography.

In contrast, the vast majority of misclassified character images with confidence levels less than 33% (i.e., Figure 7c) in our sample were blurred, if not simply impossible to read by our research team's qualified experts. As shown in Figure 8, we visually observe that images with higher levels of blurriness tend to have lower confidence scores in the RESNET model, suggesting that future experiments may benefit significantly from an image dataset thresholded on image blurriness. This observation suggests that the original digitization process, the cropping algorithm, or the consensus algorithm that was employed to create the dataset encountered a failure. In other words, our observation suggests that it was not the RESNET and CNN models that encountered a misclassification error, but the prior pipeline stage that was used to produce the dataset.

Based on this preliminary analysis, we estimate that at least 80% of the misclassifications in the RESNET model can be attributed to labeling errors, stemming from any one of the pre-processing stages (e.g., cropping, consensus assignment, or imperfect annotations) shown in Figure 2. Based on this brief manual examination of this data, only about 1% Alpha character images were incorrectly classified by the RESNET model. There were 1,125 Alpha character images that were misclassified by the RESNET model, but were not misclassified by the CNN model. When we examine these characters, we find about 20% of the characters with confidence greater than 66% were incorrectly classified. For the misclassifications situated between 33% and 66% range of confidence, about 5% of the actual Alpha characters were incorrectly classified.

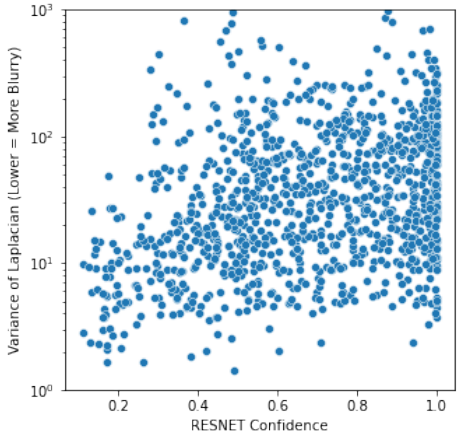
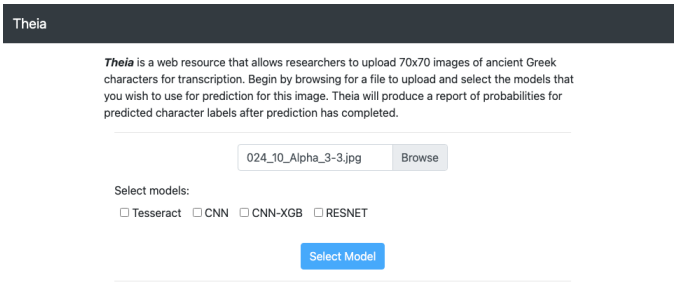


Fig. 8: Blurriness and confidence for all misidentified Alpha images from RESNET and CNN−BASE in the AL−ALL dataset.



(a) The base interface in which users upload a character image and make a selection for approaches to character recognition.

Analyzed Image:

	Tesseract	CNN
Alpha	0%	91.258%
Beta	0%	0%
Chi	0%	0%
Delta	0%	0%
Epsilon	0%	0%
Eta	0%	0%
Gamma	0%	0%
Iota	0%	0%
Kappa	0%	0%
Lambda	0%	8.741%
Mu	0%	0%
Nu	0%	0%
Omega	0%	0%
Omicron	0%	0%
Phi	0%	0%
Pi	100%	0%
Psi	0%	0%
Rho	0%	0%
Sigma	0%	0%
Tau	0%	0%
Theta	0%	0%
Upsilon	0%	0%
Xi	0%	0%
Zeta	0%	0%
Best	Pi	Alpha

(b) An example of Theia's classification output.

Fig. 9: An overview of Theia's interface.

VI. Theia: A Web UI for Character Classification

Our findings suggest that each of our three machine learning models substantially outperform the Tesseract OCR engine. Here, we introduce *Theia*, a static web interface that allows members of the research community to easily and intuitively reproduce analysis on our approaches. The interface is specifically designed for nonexperts (e.g., in the humanities) who lack the necessary technical experience to train and evaluate machine learning approaches. Theia was built using the Tensorflow.js and Tesseract.js libraries, both of which use client-side resources (i.e., GPU using WebGL compute shaders) to load learning modules and perform model-related computation. Theia's front-end centric design facilitates an affordable hosting model for the academic and research community by requiring only a standard HTTP server (i.e., for statically hosted content) to function. Further, Tensorflow.js maintains embedded support for importing re-usable learning models such that future research teams can engineer their own models (e.g., in other language formats), which could be later incorporated into and hosted within the Theia system. The Theia system can be found on the project webpage[3].

A. User Experience and Web Interface

Theia aims to enable nonexperts with the ability to identify which of our trained models are most appropriate for their own project. Theia could, for example, allow a team of archaeologists to explore the applicability of our models to a dataset of digitzed ancient Greek manuscripts that are the product of a digitization initiative that used a unique digitization technique (e.g., that is dissimilar to the technique used to digitize the Oxyrhynchus papyri studied in this paper). Users can use Theia by uploading an image of appropriate size (70x70), selecting the desired models to use for classification (e.g. Tesseract, `CNN-BASE`, `CNN-XGB`, or `RESNET`), and clicking the "Transcribe" button to send the image to the models for classification. After the models have completed their classification procedure, the image will be rendered to the user alongside a list of possible character labels and their associated classification probabilities. Note that the Tesseract model may return multiple characters, so there may appear to be more than one "best" transcription. By engaging in this process iteratively, users are capable of determining which model performs best under their specific circumstances. An overview of Theia's user experience is shown in Figure 9.

VII. Discussion

Our work demonstrates that learning methods can be usefully applied to the task of identifying handwritten characters in digitized images of ancient manuscripts. In this study's context, we designed, implemented, optimized, and explored the classification effectiveness of three unique approaches for identifying ancient Greek characters in manuscripts from the Oxyrhynchus papyri collection. Our comparative analysis with the Tesseract OCR engine suggests that each of our trained models were significantly more accurate in classifying all possible characters images across the ancient Greek alphabet.

An important consideration for interpreting our findings is the fundamental basis in which the Tesseract engine itself was trained. Like our three learning approaches, Tesseract utilizes an embedded learning approach (i.e., LSTM) to facilitate its optical character recognition [45]. Our decision to use Tesseract as a baseline measure of performance was motivated by its widespread utility as an out-of-the-box tool for effective character recognition. A follow-up study could explore training Tesseract with the `AL-ALL` and `AL-PUB`

[3]https://utk-pairs.github.io/theia/

datasets to create a custom new Tesseract OCR model, which could also be made available via Theia. Such a study would help draw finer conclusions about the performance trade-offs between the approaches evaluated in our work.

Alongside our comparative analysis, our research introduces new insights into the applicability of our learning approaches to scenarios that involve imperfect data. A wealth of research in crowdsourcing and machine learning research have given ample attention to the development of computational techniques for identifying reliable annotators, filtering out unreliable sources of data, and more generally, improving the quality of crowdsourced data [46], [47], [14]. In conjunction with new aggregation frameworks for learning from imperfect annotation data [48], we find evidence that suggests that our modelling approaches are surprisingly resilient to imperfect data. Specifically, our audit of misclassifications in Section V-C highlights that our learning approaches encounter misclassification errors with character images that may have been labeled by human annotators incorrectly, algorithmically assigned an incorrect consensus letter, or simply algorithmically cropped in an imperfect fashion. Further, we find that many misclassifications may stem from the dataset's initial digitization as described in Figure 2. Our audit introduces a new frontier for further exploring the effect of these "error-induced" misclassifications and how their exclusion from the AL-ALL and AL-PUB datasets may affect model performance.

Lastly, our findings set a compelling benchmark for future research at the intersection of character recognition, citizen science, and machine learning for cultural heritage contexts. Historically, research initiatives for manuscript transcription retain their data as a by-product of platform policy agreements. By making the AL-PUB dataset publicly available and introducing *Theia* into the research community, we believe that we've taken the necessary steps toward establishing a change of attitude in sharing data, models, and resources among the research communities that work at this intersection. Researchers can, for example, extend Theia to datasets of alternative languages of interest and incorporate new learning models for widespread use among the community. With the AL-PUB dataset, machine learning researchers now have a readily-available alternative to the MNIST dataset [1] that has been exhaustively used for nearly three decades of research. In general, these contributions facilitate the re-use, replication, and extension of our research in the interest of generating new advances for computing and the humanities alike.

VIII. Conclusion

In this paper, we explored the performance of state-of-the-art optical character recognition tools for print and learning models engineered with state-of-the-art machine learning toolkits trained on handwritten inputs. Using Tesseract OCR as a baseline, we build, optimize, and evaluate three types of convolutional neural networks that are trained on the AL-ALL and AL-PUB[4] datasets, a collection of images of handwritten ancient Greek characters that were labeled by volunteers through the Ancient Lives online citizen science project. We find our best-performing machine learning model to be 92.57% accurate compared to Tesseract OCR's 11.15%. Following our analysis, we present a brief examination of our models' shortcomings, introduce the publicly-available AL-PUB dataset, and, describe *Theia*, a web-based tool that democratizes our machine learning models for public use. We conclude by discussing the promise of our findings for advancing research at the intersection of machine learning, manuscript transcription, and the digital humanities.

[4]https://data.cs.mtsu.edu/al-pub

Acknowledgment

This research is made possible by the thousands of Zooniverse volunteers who participated in the Ancient Lives project over the past decade. We recognize these volunteers and thank them for their efforts in spurring advances not only across the humanities, but also, now, the sciences. We also thank the Imaging Papyri Project at the University of Oxford for providing access to the digitized manuscript images as well as the Egyptian Exploration Society for providing access to the Oxyrhynchus Papyri. This research was partially funded by the Andrew W. Mellon Foundation and The Chellgren Center for Undergraduate Excellence.

References

[1] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012. 1, 9

[2] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*, 1990, pp. 396–404. 1

[3] Y. LeCun *et al.*, "Generalization and network design strategies," *Connectionism in perspective*, vol. 19, pp. 143–155, 1989. 1

[4] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "Emnist: Extending mnist to handwritten letters," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 2921–2926. 1, 2

[5] A. Pal and D. Singh, "Handwritten english character recognition using neural network," *International Journal of Computer Science & Communication*, vol. 1, no. 2, pp. 141–144, 2010. 1

[6] M. Diem and R. Sablatnig, "Recognizing characters of ancient manuscripts," in *Computer Vision and Image Analysis of Art*, vol. 7531. International Society for Optics and Photonics, 2010, p. 753106. 1

[7] N. Reggiani, *Digital Papyrology I. Methods, Tools and Trends*. De Gruyter, 2017. 1

[8] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik, "A multi-view embedding space for modeling internet images, tags, and their semantics," *International journal of computer vision*, vol. 106, no. 2, pp. 210–233, 2014. 1

[9] D. F. Nettleton, A. Orriols-Puig, and A. Fornells, "A study of the effect of different types of noise on the precision of supervised learning techniques," *Artificial intelligence review*, vol. 33, no. 4, pp. 275–306, 2010. 1, 2

[10] M. Pechenizkiy, A. Tsymbal, S. Puuronen, and O. Pechenizkiy, "Class noise and supervised learning in medical domains: The effect of feature extraction," in *19th IEEE symposium on computer-based medical systems (CBMS'06)*. IEEE, 2006, pp. 708–713. 1, 2

[11] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," *Artificial intelligence review*, vol. 22, no. 3, pp. 177–210, 2004. 1, 2

[12] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 138–156, 2006. 1

[13] C.-M. Teng, "A comparison of noise handling techniques." in *FLAIRS Conference*, 2001, pp. 269–273. 1

[14] A. C. Williams, J. Goh, C. G. Willis, A. M. Ellison, J. H. Brusuelas, C. C. Davis, and E. Law, "Deja vu: Characterizing worker reliability using task consistency." in *HCOMP*, 2017, pp. 197–205. 1, 2, 9

[15] X. Zhur and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," 2002. 1

[16] T. E. De Campos, B. R. Babu, M. Varma *et al.*, "Character recognition in natural images." *VISAPP (2)*, vol. 7, 2009. 2

[17] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015. 2

[18] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, "Online and offline handwritten chinese character recognition: benchmarking on new databases," *Pattern Recognition*, vol. 46, no. 1, pp. 155–162, 2013. 2

[19] Y. Hong, Q. Li, J. Jiang, and Z. Tu, "Learning a mixture of sparse distance metrics for classification and dimensionality reduction," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 906–913. 2

[20] M. Thoma, "The hasyv2 dataset," *arXiv preprint arXiv:1701.08380*, 2017. 2

[21] M. Karki, Q. Liu, R. DiBiano, S. Basu, and S. Mukhopadhyay, "Pixel-level reconstruction and classification for noisy handwritten bangla characters," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2018, pp. 511–516. 2

[22] L. Blaser, "Old weather: approaching collections from a different angle," *Crowdsourcing our cultural heritage*, pp. 45–56, 2014. 2

[23] R. Grayson, "A life in the trenches? the use of operation war diary and crowdsourcing methods to provide an understanding of the british army's day-to-day life on the western front," *British Journal for Military History*, vol. 2, no. 2, 2016. 2

[24] A. Mao, E. Kamar, Y. Chen, E. Horvitz, M. E. Schwamb, C. J. Lintott, and A. M. Smith, "Volunteering versus work for pay: Incentives and tradeoffs in crowdsourcing," in *First AAAI conference on human computation and crowdsourcing*. Citeseer, 2013. 2

[25] T. Causer and M. Terras, "Many hands make light work. many hands together make merry work': Transcribe bentham and crowdsourcing manuscript collections," *Crowdsourcing our cultural heritage*, pp. 57–88, 2014. 2

[26] A. C. Williams, H. D. Carroll, J. F. Wallin, J. Brusuelas, L. Fortson, A.-F. Lamblin, and H. Yu, "Identification of ancient greek papyrus fragments using genetic sequence alignment algorithms," in *2014 IEEE 10th international conference on e-science*, vol. 2. IEEE, 2014, pp. 5–10. 2, 3

[27] A. C. Williams, J. F. Wallin, H. Yu, M. Perale, H. D. Carroll, A.-F. Lamblin, L. Fortson, D. Obbink, C. J. Lintott, and J. H. Brusuelas, "A computational pipeline for crowdsourced transcriptions of ancient greek papyrus fragments," in *2014 IEEE International Conference on Big Data (Big Data)*. IEEE, 2014, pp. 100–105. 2, 3

[28] R. Simpson, K. R. Page, and D. De Roure, "Zooniverse: observing the world's largest citizen science platform," in *Proceedings of the 23rd International Conference on World Wide Web - WWW '14 Companion*. New York, New York, USA: ACM Press, 2014, pp. 1049–1054. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2567948.2579215 2, 3

[29] J. Ortiz Baco, A. Guzman, and A. A. Palacios, "Fromthepage collection owner user study report," *Enabling and Reusing Multilingual Citizen Contributions in the Archival Record-NEH Grant Documentation*, 2020. 2

[30] C. Bonacchi, A. Bevan, A. Keinan-Schoonbaert, D. Pett, and J. Wexler, "Participation in heritage crowdsourcing," *Museum Management and Curatorship*, vol. 34, no. 2, pp. 166–182, 2019. 2

[31] A. K. Bowman, "Oxyrhynchus in the early fourth century:" municipalization" and prosperity," *The Bulletin of the American Society of Papyrologists*, pp. 31–40, 2008. 3

[32] J. H. Brusuelas, "Engaging greek: Ancient lives," *Digital Classics Outside the Echo-Chamber: Teaching, Knowledge*, pp. 187–204, 2016. 3

[33] A. K. Bowman, R. A. Coles, N. Gonis, D. Obbink, and P. J. Parsons, *Oxyrhynchus: a City and its Texts*. Egypt Exploration Society 93, 2007. 3

[34] R. Smith, "An overview of the tesseract ocr engine," in *Ninth international conference on document analysis and recognition (ICDAR 2007)*, vol. 2. IEEE, 2007, pp. 629–633. 4

[35] R. W. Smith, "History of the tesseract ocr engine: what worked and what didn't," in *Document Recognition and Retrieval XX*, vol. 8658. International Society for Optics and Photonics, 2013, p. 865802. 4

[36] G. Franzini, F. Zampedri, M. Passarotti, F. Mambrini, and G. Moretti, "Græcissâre: Ancient greek loanwords in the lila knowledge base of linguistic resources for latin," in *Seventh Italian Conference on Computational Linguistics*. CEUR-WS. org, 2020, pp. 1–6. 4

[37] N. White, "Training tesseract for ancient greek ocr," *Eiiruzov*, no. 28–29, 2012. 4

[38] C. Patel, A. Patel, and D. Patel, "Optical character recognition by open source ocr tool tesseract: A case study," *International Journal of Computer Applications*, vol. 55, no. 10, pp. 50–56, 2012. 4

[39] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional neural network committees for handwritten character classification," in *2011 International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 1135–1139. 4

[40] T. Guo, J. Dong, H. Li, and Y. Gao, "Simple convolutional neural network on image classification," in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(*. IEEE, 2017, pp. 721–724. 4

[41] R. Vaidya, D. Trivedi, S. Satra, and M. Pimpale, "Handwritten character recognition using deep-learning," in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*. IEEE, 2018, pp. 772–775. 4

[42] S. Thongsuwan, S. Jaiyen, A. Padcharoen, and P. Agarwal, "Convxgb: A new deep learning model for classification problems based on cnn and xgboost," *Nuclear Engineering and Technology*, vol. 53, pp. 522–531, 2020. 4

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. 4

[44] N. M. Aszemi and P. Dominic, "Hyperparameter optimization in convolutional neural network using genetic algorithms," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 6, pp. 269–278, 2019. 5

[45] T. M. Breuel, "High performance text recognition using a hybrid convolutional-lstm implementation," in *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 11–16. 8

[46] A. Das Sarma, A. Parameswaran, and J. Widom, "Towards globally optimal crowdsourcing quality management: The uniform worker setting," in *Proceedings of the 2016 International Conference on Management of Data*. ACM, 2016, pp. 47–62. 9

[47] H. J. Jung and M. Lease, "Improving consensus accuracy via z-score and weighted voting." in *Proceedings of the 2011 AAAI Workshop on Human Computation*, 2011. 9

[48] E. A. Platanios, M. Al-Shedivat, E. Xing, and T. Mitchell, "Learning from imperfect annotations," 2020. 9