arXiv:2210.16380v1 [cs.CV] 28 Oct 2022

# An Approach for Noisy, Crowdsourced Datasets Utilizing Ensemble Modeling, 'Human Softmax' Distributions, and Entropic Measures of Uncertainty

**Graham West,** [1] **Matthew I. Swindall,** [1] **Ben Keener,** [2] **Timothy Player,** [4]
**Alex C. Williams,** [3] **James H. Brusuelas,** [2] **John F. Wallin** [1]

[1] Middle Tennessee State University
[2] University of Kentucky
[3] Amazon
[4] University of Tennessee, Knoxville

**Abstract**

Noisy, crowdsourced image datasets prove challenging, even for the best neural networks. Two issues which complicate classification on such datasets are class imbalance and ground-truth uncertainty in labeling. The AL-ALL and AL-PUB datasets—consisting of tightly cropped, individual characters from images of ancient Greek papyri—are strongly affected by both issues. The application of ensemble modeling to such a dataset can help identify images where the ground-truth is questionable and quantify the trustworthiness of those samples. We apply stacked generalization consisting of nearly identical ResNets: one utilizing cross-entropy (CXE) and the other Kullback-Liebler Divergence (KLD). The CXE network uses standard labeling drawn from the crowdsourced consensus. In contrast, the KLD network uses probabilistic labeling for each image derived from the distribution of crowdsourced annotations. We refer to this labeling as the Human Softmax (HSM) distribution. For our ensemble model, we apply a $k$-nearest neighbors model to the outputs of the CXE and KLD networks. Individually, the ResNet models have approximately 93% accuracy, while the ensemble model achieves an accuracy of >95%. We also perform an analysis of the Shannon entropy of the various models' output distributions to measure classification uncertainty.

## 1 Introduction

One of the greatest challenges in the domain of machine learning is the acquisition of useful datasets. Three mainstream approaches include:

1. Professional Annotation Services
2. Crowdsourcing Initiatives
3. Synthetic Dataset Generation

In many cases, approaches 1 and 3 can be costly endeavors while approach 2 can require years of effort. In all three cases, further challenges arise from both the target domains and the different processes for labeling the resulting data. Datasets developed by these methods are often described as 'noisy' due to the lack of clarity in ground-truth. Generative Adversarial Networks (GANs) can produce astonishingly realistic synthetic image datasets (Swindall et al. 2022), but for large scale projects it is impractical to verify that every feature and label are correct. Professional annotation services such as Amazon Mechanical Turk (Sorokin and Forsyth 2008) sources dataset labeling from human annotators who are unlikely to be domain experts. Crowdsourcing initiatives encounter the same challenge, as most annotations are performed by untrained volunteers. In each of these cases there is usually a lack of certainty in the ground-truth of the produced dataset.

In the field of papyrology (the study of ancient Greek papyri), these challenges are compounded by the lack of quality image datasets and a shortage of experts who are qualified to annotate them. Efforts to produce a crowdsourced image dataset from such manuscripts, such as AL-ALL, and AL-PUB (Swindall et al. 2021; Williams et al. 2014) have shown that, even though the ground-truth in labeling may be questionable, the utilization of annotator consensus can produce datasets that can be accurately modeled using deep learning approaches. To build on these efforts, we propose a stacked generalization method comprised of two ResNets trained on the AL-ALL dataset with identical architectures but different loss functions and labeling schemes. We then explore the Shannon entropy of the various models' Softmax outputs and the utility of entropy as a measure of classification accuracy. Our results demonstrate that the proposed ensemble method yields greater accuracy than either of the individual models and highlights the ability of the Shannon entropy to quantify ground-truth uncertainty of each image's crowdsourced annotation distribution within the dataset.

### 1.1 Deep Learning and Handwritten Character Recognition

While deep learning models can classify characters with astounding accuracy, ancient handwritten manuscripts still pose many challenges. Existing character recognition engines have been mostly trained on modern, printed documents and perform poorly for handwritten texts. In the case of highly damaged, ancient Greek papyri, the failure of such engines demonstrates the need for custom-trained models (Swindall et al. 2021).

### 1.2 The Ancient Lives Project

The Ancient Lives Project was a web-based crowdsourcing initiative that allowed volunteers to transcribe digital images of ancient papyrus fragments from 2011 until 2018 (Williams et al. 2014). These manuscripts consist of handwriting on papyrus, in various states of preservation, exca-

Figure 1: An Example of an Oxyrhynchus Papyrus Fragment

vated from rubbish dumps near the ancient Egyptian city of Oxyrhynchus (Bowman et al. 2007). This project resulted in millions of annotations from images of papyrus fragments, such as the one shown in Figure 1. A consensus algorithm was applied to these annotations to produce approximate locations of individual characters within the document images, as well as a consensus labels for each identified character.

### 1.3 Datasets: AL-ALL and AL-PUB

The consensus data from the Ancient Lives Project was used to create two crowdsourced datasets: AL-ALL (containing 399,421 images from published and unpublished papyri) and AL-PUB (containing 195,683 images from only published papyri). Both datasets consist of tightly cropped images of individual Greek characters from the annotated papyri images. Figure 2 shows example images from AL-PUB, one for each character in the standard Greek alphabet.

Although the datasets are full of excellent character instances, the data are "noisy." Unlike datasets such as MNIST (Deng 2012), AL-ALL and AL-PUB were created from images containing holes, rips, missing segments, and faded ink. As shown in Figure 3, many character images are illegible or incomplete. Consequently, this compounds ground-truth uncertainty in the labeling of such images. Furthermore, initiatives like the Ancient Lives Project must rely on untrained individuals to record both the correct classification and a reasonably accurate location of the character within the manuscript image, leading to supplemental noise in labeling.

Additionally, there is significant sample bias inherent to these datasets due to multiple factors. The primary source of sample bias is the lack of instances of certain characters in the source manuscripts themselves. This bias is compounded by the tendency of human annotators to "chase characters" i.e., preferentially annotate specific characters. Table 1 lists the sample sizes for each character in AL-ALL. While most characters have many thousands of samples, there are several which have a relatively small amount. The most significant of these imbalanced outliers is $\sigma$, with only 62 samples. Due to its small sample size, $\sigma$ is the most challenging character to predict accurately and our models perform very

Figure 2: Examples of Character Images from the AL-PUB Dataset

Figure 3: Characters From Damaged Papyri

poorly on this character. As some of the fragment images used to create AL-ALL are not yet in the public domain, Al-PUB was created to share with the public. Though AL-PUB is a smaller subset of AL-ALL, previous work suggests minimal model performance impairment when training with AL-PUB versus AL-ALL (Swindall et al. 2021). However, to maximize the benefits of the larger dataset, our image classification models were trained on AL-ALL.

### 1.4 Human Softmax (HSM)

Inspired by the Softmax output layer in most multi-class classification models, we sought to design a similar probabilistic representation of the annotation data from the Ancient Lives Project. Annotations for each image were converted into arrays, consisting of the number of annotations for each class. These values are then normalized by feature scaling as shown in equation 1

$$x'_i = \frac{x_i}{\sum_{j=1}^{M} x_j} \tag{1}$$

where $M = 24$ classes, $x_i$ is the number of human annotations for class $i$ and $x'_i$ is its normalized value. This yields a representation similar to a Softmax output layer. We call this distribution the Human Softmax, or HSM. An example of the HSM distribution for an individual image is shown in Figure 4. This is an excellent illustration of the difficult nature of AL-ALL as the ground-truth is unclear. This specific character, shown in Figure 8, is labeled as Gamma ($\Gamma$) based on consensus. However, many volunteers thought the character was Psi ($\Psi$). Even experts in papy.rology could debate whether this image is indeed a Gamma.

## 2 Methods

The overarching goal of this work is to build a pipeline for improving noisy, challenging image datasets. In this sec-

| Character | Count | Character | Count |
|---|---|---|---|
| **Alpha** $(A, \alpha)$ | 42,546 | **Nu** $(N, \nu)$ | 44,910 |
| **Beta** $(B, \beta)$ | 2,534 | **Xi** $(\Xi, \xi)$ | 1,201 |
| **Gamma** $(\Gamma, \gamma)$ | 6,907 | **Omicron** $(O, o)$ | 46,344 |
| **Delta** $(\Delta, \delta)$ | 11,717 | **Pi** $(\Pi, \pi)$ | 17,114 |
| **Epsilon** $(E, \epsilon)$ | 31,584 | **Rho** $(P, \rho)$ | 20,450 |
| **Zeta** $(Z, \zeta)$ | 1,425 | **Sigma** $(\Sigma, \sigma)$ | 62 |
| **Eta** $(H, \eta)$ | 15,064 | **Tau** $(T, \tau)$ | 32,045 |
| **Theta** $(\Theta, \theta)$ | 7,575 | **Upsilon** $(Y, \upsilon)$ | 15,762 |
| **Iota** $(I, \iota)$ | 25,595 | **Phi** $(\Phi, \phi)$ | 6,063 |
| **Kappa** $(K, \kappa)$ | 17,937 | **Chi** $(\mathrm{X}, \chi)$ | 9,156 |
| **Lambda** $(\Lambda, \lambda)$ | 13,253 | **Psi** $(\Psi, \psi)$ | 904 |
| **Mu** $(M, \mu)$ | 13,227 | **Omega** $(\Omega, \omega)$ | 16,046 |

Table 1: Counts for each letter in the Ancient Lives dataset.

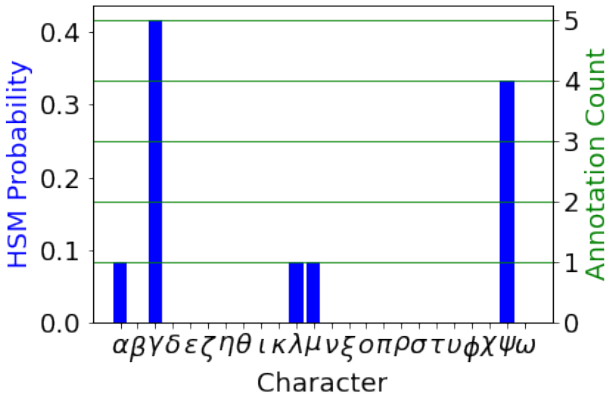


Figure 4: Human Softmax Distribution (HSM) for Image Shown in Figure 8

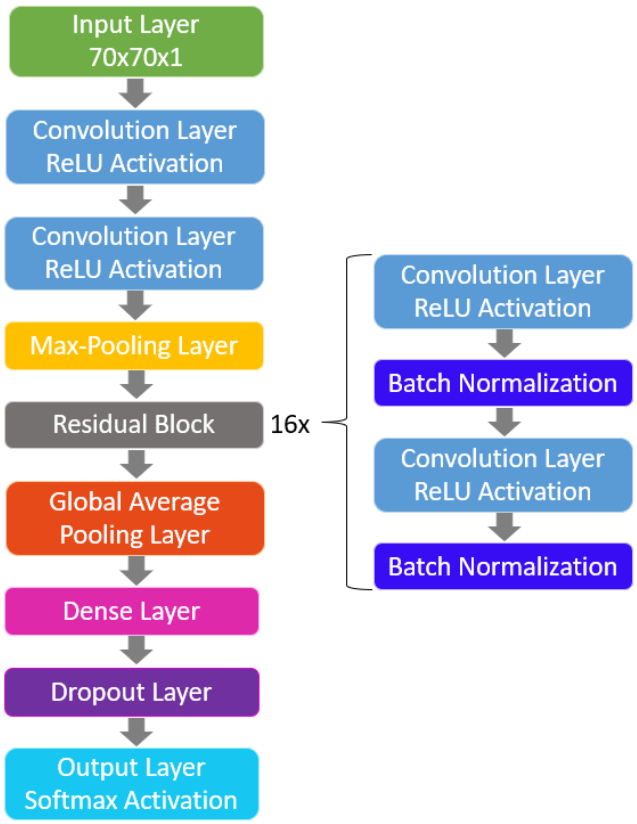


Figure 5: ResNet Architecture

tion, we discuss the various methods used in this study. We describe the architecture of the ResNets, their labeling schemes, and their loss functions. We then describe our approach for performing stacked generalization to combine the ResNets' outputs to create input features for a $k$-nearest neighbors model. Lastly, we explore the evaluation metrics which were used to monitor the training progress of the ResNets.

An important semantic distinction must be addressed before further discussion. As the ground-truth can often be ambiguous for crowd-sourced datasets, it is necessary to assume the consensus labels, based on human annotation, to be the ground-truth. Thus, when we indicate that a particular image was "correctly classified" by a given model, we are stating that the model prediction agrees with the consensus annotation. Conversely, when we state an image is "incorrectly classified," we are indicating that the model disagrees with consensus.

## 2.1 Deep Learning Architecture

The core of the proposed ensemble model consists of two, nearly identical architectures. The ResNet, outlined in Figure 5, is a fairly standard convolutional residual network (He et al. 2016). The early layers in the model consist of two convolution layers with ReLU activation, followed by a max-pooling layer and 16 residual blocks. The residual blocks consist of two cycles of convolution and batch normalization. After the residual block, the model finishes with a global average pooling layer, a single dense layer, a dropout layer, and finally an output layer. These models differ only in the class labels and loss functions. The architecture is based on the models utilized in (Swindall et al. 2021). For both models, the input features are identical: the raw images. The first model, CXE-ResNet, takes standard, single value numerical class labels while the second model, KLD-ResNet, takes the HSM distribution from the Ancient Lives Project data as labels.

The ResNets, written in Tensorflow 2.3 with GPU acceleration, were trained on a Linux system consisting of 2 Intel Xeon E5-2687W Processors and 2 NVIDIA GeForce RTX 2080 Ti CUDA cores. In the interest of limiting the differences between the two ResNets, the same random seed was used for both models, though this likely had minimal effect on training or inference. Recent work (Morin and Willetts 2020) suggests that GPU non-determinism dominates the effects of seed randomization.

**Loss functions** We trained ResNets with two different loss functions: sparse categorical cross-entropy (CXE) and Kullback-Leibler divergence (KLD). Both of these functions

are based on the Shannon entropy $H$ (C. E. Shannon 1949) given below in Equation 2.

$$H(p) = \mathbf{E}[-log(p)] = -\sum_{i=1}^{M} p_i log(p_i) \tag{2}$$

Here, $p = (p_1, \cdots, p_M)$ is some probability distribution defined over $M$ classes. This function (which takes an entire probability distribution as input) returns the amount of "uncertainty" present in the distribution. Thus, for distributions which are very narrowly peaked, the entropy will be low, while for distributions which are more spread out and uniform, the entropy will be high. The full range of values of the Shannon entropy is given by $H(p) \in [0, \log(M)]$ for all possible distributions $p$.

The first of our loss functions—sparse categorical cross entropy—can be defined in terms of the Shannon entropy as in Equation 3 and interpreted as the entropy of $p$ *with respect to* $q$.

$$H_q(p) = \mathbf{E}_q[-log(p_i)] = -\sum_{i=1}^{M} q_i log(p_i) \tag{3}$$

where $p$ is the model's output distribution and $q$ the "target" distribution.

In our case, the consensus labeling scheme results in the target taking on a delta distribution, where there is a probability of one for the correct class and zero for the rest (giving the distribution zero entropy). In order to calculate the CXE for the entire data set, one simply adds the values $H_q(p)$ for each image.

$$\text{CXE} = \sum_{j=1}^{N} H_{q^j}(p^j) \tag{4}$$

Here, $p^j, q^j$ are the model and target distributions of image $j$ (of which there are $N$).

The second loss function—*Kullback-Leibler divergence* (S. Kullback 1951)—is defined in Equation 5.

$$D_{KL}(p||q) = H_q(p) - H(p) \tag{5}$$

This function is simply the difference between the cross entropy and the standard entropy. This value is always non-negative since $H_q(p) \geq H(p)$. Also, as with the CXE, one calculates the KLD for the entire dataset by summing $D_{KL}(p||q)$ for the individual images. However, instead of using the consensus class as the label, we use the entire HSM as our target distribution. Due to the fact that these distributions are rarely delta distributions, they tend to have a non-zero entropy.

### 2.2 Ensemble Model: Stacked Generalization and $k$-Nearest Neighbors

As will be shown below, though both ResNets performed well, there is a sizable subset of the data which one network classified correctly while the other was incorrect. Because of this disagreement between the ResNets, it is possible to construct an ensemble model which uses the two models' outputs to achieve an accuracy greater than that of the two individual networks (Goodfellow, Bengio, and Courville 2016). We employ the technique of "stacked generalization" (Wolpert 1992) whereby the output distributions of the two ResNets are used as inputs to a new model. Using two ResNets with different loss functions and labeling schemes decreases the amount of correlation between the models, giving us a more robust ensemble model. We use a $k$-nearest neighbors model (hereafter KNN) with $k = 50$ for this secondary model. To do this, we let $C^i_j, K^i_j$ be the probability (taken from the CXE- and KLD-ResNet outputs, respectively) that the $i$-the image belongs to the $j$-th class. We can then create a new set of input data $X$ for the KNN model via Equation 6.

$$\begin{aligned} X^{1,\cdots,N}_{1,\cdots,M} &= C^{1,\cdots,N}_{1,\cdots,M} \\ X^{1,\cdots,N}_{M+1,\cdots,2M} &= K^{1,\cdots,N}_{1,\cdots,M} \end{aligned} \tag{6}$$

Note that the lower index of $X$ takes on $2M$ values since there are two models each with $M = 24$ classes whose output probabilities must be incorporated.

Analogous to the Softmax output layer of the ResNets, the KNN produces an output distribution of classification probabilities which is based on the fraction of neighbors of different classes. This allows us to perform the same entropic analyses on all three models.

### 2.3 Evaluation Metrics

While the CXE-ResNet utilizes the standard accuracy metric for evaluation, the approach used by the KLD-ResNet is atypical of deep learning approaches. This is by nature of the labeling scheme used. Thus, Mean-Absolute-Error (MAE) was utilized as the primary metric. This posed a challenge when comparing the results from training the two models. The most straight-forward way of comparing accuracy for our models is to run inference on the entire dataset, including both training and validation data, and calculating the accuracy of model predictions. For consistency, we continue this approach in analysis of the ensemble model.

## 3 Numerical Experiments

We now move on to discuss the different numerical experiments performed on the data. We will begin with a discussion of the experimental setup and then proceed to compare the results of the different models tested. We will also provide some analysis of these results based on the Shannon entropy of the models' Softmax predictions. For brevity, we only share here a limited number of plots consisting of the results for the entire dataset and results for the single character $\alpha$. Hundreds of plots were produced showing results for all classes. These plots, as well as the saved models, will be made available at our website —*Website Redacted For Anonymity*—

### 3.1 Experimental setup

For measuring the accuracy of our models, we make use of the precision and recall statistics, given in Equation 7

$$\begin{aligned} \text{Precision} &= TP/(TP + FP) \\ \text{Recall} &= TP/(TP + FN) \end{aligned} \tag{7}$$

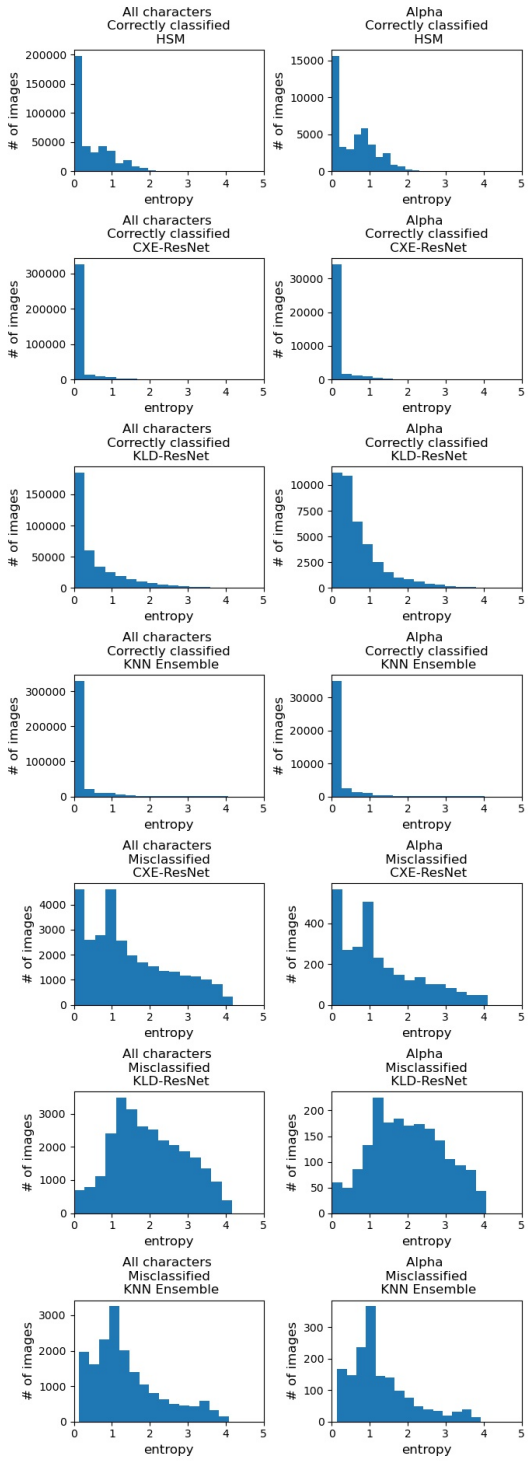


Figure 6: Histogram of entropies for the HSM and the three models' output distributions for both the entire data set and solely the character $\alpha$. We split the model histograms into two portions: 1) images whose model classification agreed with the human consensus and 2) those which disagreed.

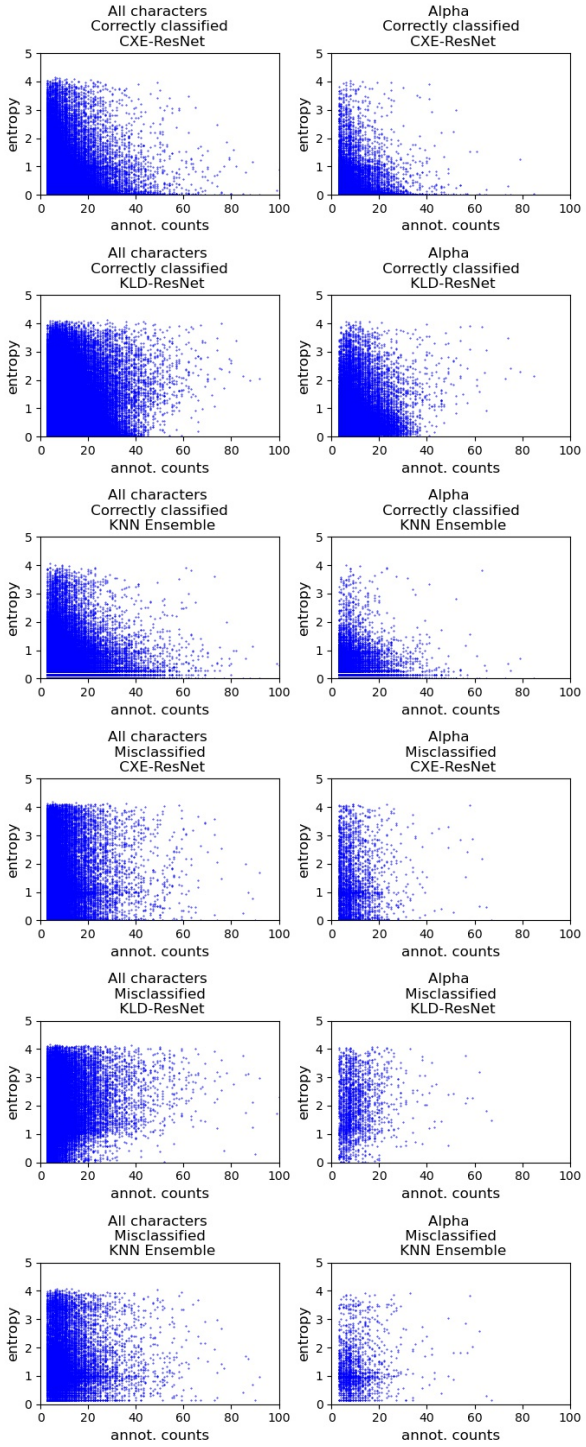


Figure 7: Plots of the entropies of the three models' output distributions versus the number of human annotations for both the entire data set and solely the character $\alpha$. We split the plots into two portions: 1) images whose model classification agreed with the human consensus and 2) those which disagreed..

Figure 8: AL-PUB Gamma ($\Gamma$) Example. File Name: Z_POxy.v0015.n1805.a.01 _135790_216_Gamma_5-12.jpg

| | | KLD | |
|---|---|---|---|
| | | Cor | Inc |
| CXE | Cor | 353549 | 16407 |
| | Inc | 18594 | 10871 |

Table 2: Number of images correctly/incorrectly classified by the CXE- and KLD-ResNets.

where $TP$ is the number of true positives, $FP$ is the number of true negatives, and $FN$ is the number of false negatives.

### 3.2 Results

Table 2 shows the number of correct and incorrect classifications made by the CXE- and KLD-ResNet models. While both models correctly classify approximately 93% of the images (see below for a more thorough look at model accuracy), the off-diagonal elements in the table show that there is a sizeable fraction of images which one model classified correctly while the other classified incorrectly. As stated earlier, stacked generalization can provide a boost in accuracy in such a case, where there is enough disagreement between two well-performing models that the best of both can be incorporated.

Table 3 and Figure 9 contain the precision and recall for the CXE-ResNets, KLD-ResNets, and KNN ensemble models per character and for the entire data set. Overall, the models achieved accuracies of 92.6%, 93.2%, and 95.6%, respectively.

Note that for the majority of characters and models, the precision and recall are $>$0.9, with the KNN tending to have the highest values. Cases with lower values for these statistics tend to be characters with fewer sample images (see Figure 10). The most notable of these would be $\sigma$, for which there are only 62 samples (compared to $>$42,000 for $\alpha$). While the two ResNets had precisions of 0.500 and 0.857 for $\sigma$, respectively, the recalls where $<$0.1 for both. Even more significant is the fact that the KNN achieved both precision and recall of 0.0. This is reasonable since the number of neighbors (50) is near the number of samples (62), making the chance of a majority of $\sigma$ neighbors quite low.

### 3.3 Entropy Analysis

We also performed entropic analyses on the HSM, ResNets, and ensemble models' output distributions. Figure 6 shows histograms for the entropy of each of these distributions for both the entire data set and for just the character $\alpha$. The plots are split into two, with one plot containing only those images

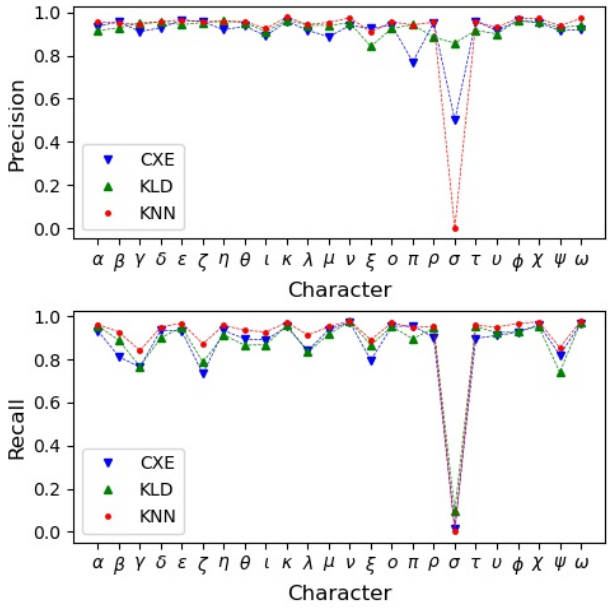


Figure 9: Per character precision and recall for each of the three models. Note that the KNN ensemble model is consistently higher than the individual models for most characters.

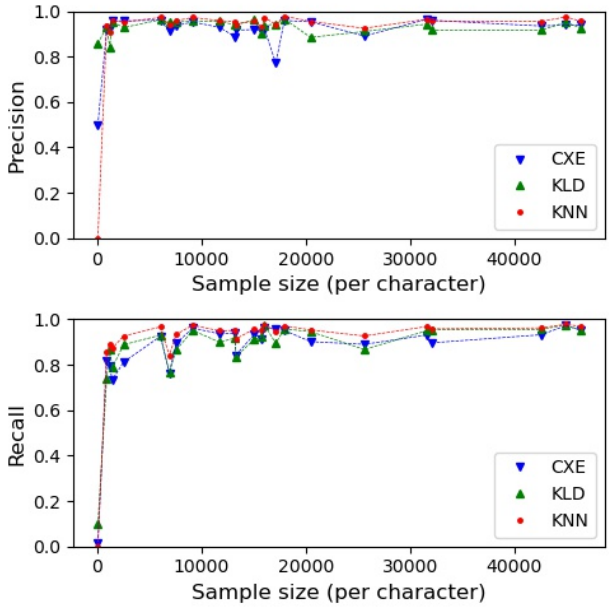


Figure 10: Precision and recall of all three models versus per character sample size. Notice the drop in recall for characters with smaller sample sizes

| | CXE | | KLD | | KNN | |
|---|---|---|---|---|---|---|
| | pre. | rec. | pre. | rec. | pre. | rec. |
| total | 0.928 | 0.926 | 0.932 | 0.932 | 0.956 | 0.956 |
| $\alpha$ | 0.935 | 0.932 | 0.916 | 0.956 | 0.954 | 0.962 |
| $\beta$ | 0.957 | 0.812 | 0.928 | 0.888 | 0.952 | 0.927 |
| $\gamma$ | 0.910 | 0.764 | 0.951 | 0.767 | 0.942 | 0.841 |
| $\delta$ | 0.928 | 0.936 | 0.955 | 0.900 | 0.958 | 0.950 |
| $\epsilon$ | 0.964 | 0.931 | 0.943 | 0.950 | 0.964 | 0.968 |
| $\zeta$ | 0.957 | 0.733 | 0.953 | 0.787 | 0.957 | 0.872 |
| $\eta$ | 0.918 | 0.934 | 0.962 | 0.913 | 0.956 | 0.958 |
| $\theta$ | 0.936 | 0.894 | 0.952 | 0.866 | 0.958 | 0.934 |
| $\iota$ | 0.891 | 0.890 | 0.910 | 0.868 | 0.924 | 0.927 |
| $\kappa$ | 0.959 | 0.951 | 0.965 | 0.958 | 0.978 | 0.971 |
| $\lambda$ | 0.917 | 0.841 | 0.940 | 0.834 | 0.943 | 0.913 |
| $\mu$ | 0.887 | 0.939 | 0.939 | 0.918 | 0.953 | 0.951 |
| $\nu$ | 0.938 | 0.972 | 0.951 | 0.975 | 0.974 | 0.978 |
| $\xi$ | 0.926 | 0.796 | 0.842 | 0.866 | 0.909 | 0.891 |
| $o$ | 0.939 | 0.952 | 0.925 | 0.952 | 0.956 | 0.969 |
| $\pi$ | 0.770 | 0.956 | 0.943 | 0.895 | 0.942 | 0.947 |
| $\rho$ | 0.953 | 0.902 | 0.884 | 0.946 | 0.955 | 0.953 |
| $\sigma$ | 0.500 | 0.016 | 0.857 | 0.097 | 0.000 | 0.000 |
| $\tau$ | 0.959 | 0.897 | 0.917 | 0.954 | 0.957 | 0.958 |
| $\upsilon$ | 0.915 | 0.911 | 0.900 | 0.922 | 0.930 | 0.949 |
| $\phi$ | 0.960 | 0.925 | 0.962 | 0.931 | 0.972 | 0.967 |
| $\chi$ | 0.952 | 0.962 | 0.955 | 0.951 | 0.972 | 0.974 |
| $\psi$ | 0.916 | 0.820 | 0.930 | 0.740 | 0.938 | 0.855 |
| $\omega$ | 0.919 | 0.966 | 0.937 | 0.974 | 0.972 | 0.977 |

Table 3: Overall and per-character precision and recall (with respect to the human consensus) for the CXE, KLD, and KNN ensemble models.

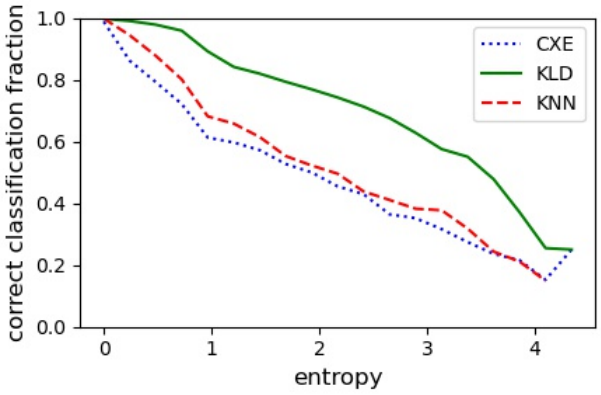


Figure 11: Plots of the fraction of correct classifications versus Softmax entropy for each model. Note that the KLD model is **NOT** better than the other models. Rather, it has consistently higher entropy values than the other models.

which were correctly classified by the respective model and the other plot containing those which were misclassified.

There are several interesting points revealed by these plots. First—and perhaps most readily noticed—is the fact that the majority of images have a low entropy in their respective output distributions, regardless of which model was used. For the HSM and correctly classified model histograms, the first bin has the greatest number of images by far. This means that for most images which are classified correctly, the model tends to be quite certain in its classification. For the misclassified model histograms, however, this is not the case (with the exception that the CXE-ResNet histograms in the first bin is approximately equal to the fourth). In fact, the tails of these histograms tend to contain the majority of samples. This means that for images which are classified incorrectly, the models tended to have less certainty in classification. Because of this discrepancy between the entropies of correct and incorrect classifications, entropy can serve not only as a good measure of classification uncertainty, but also as a predictor of classification accuracy (we discuss this further below). Images which have lower entropy will tend to be classified correctly while images with higher entropy will tend to be classified incorrectly. Figure 11 shows the fraction of correctly classified images versus entropy for the three models, illustrating the previous point well. Note, however, that the plot should not be interpreted as the KLD-ResNet significantly outperforming the other models. Rather, by nature of its labeling scheme, its Softmax distributions have a high entropy on average.

Second, note the difference in Figure 6 between the three models' entropy profiles. The KLD-ResNet tends to have higher entropy values than both the CXE-ResNet and ensemble model. The discrepancy in entropy between the CXE-ResNet and KLD-ResNet is reasonable, since the CXE-ResNet only took into account the consensus label while the KLD-ResNet used the entire human Softmax. Additionally, it is reasonable for the ensemble entropies to be quite small since a KNN model's output distribution

is merely derived from the fractions of neighbors of each class. Due to the clustering of the CXE-ResNet and KLD-ResNet output distributions (the input features for the ensemble model), the majority of samples of a given character are surrounded in feature space by other samples of the same character, inducing a low entropy in the KNN's output distribution.

Figure 7 depicts plots of the entropy of the models' output distributions for each image against the number of human annotations each image received. The most notable feature of these plots is the tendency of correctly classified images with a high annotation count to have a lower entropy than those with fewer annotations. This is reasonable when considered in conjunction with the law of large numbers. One might imagine in the limit of an infinite number of human annotations, each image would have a human Softmax which converges to some distribution, say, $P$. Consequently, the entropy $H(P)$ of $P$ would also converge. Also, as discussed above, images which are classified correctly tend to have low entropies. Therefore, as the number of human annotations for a given image which would be correctly classified increases, the entropy will tend to converge to a small value (though likely non-zero); it may, however, vary wildly when there are few annotations.

Concerning plots of the misclassified data, we do not see a consistent trend as we did with the correctly classified data. For the KLD-ResNet plots, we see that the entropy tends to take on higher values at higher annotation counts. For the CXE-ResNet and KNN plots, it is difficult to discern a clear trend.

**Entropy as a predictor for classification accuracy** As mentioned above, the entropy of a given image's output distribution (regardless of which model is used) is effective for predicting whether that image will be classified correctly by the model. To illustrate this, we trained a Support Vector Machine (SVM) to take the entropy of a model's output distribution for a given image as a single input feature and predict whether the model which produced this distribution would correctly classify the respective image. Thus, there are only two classes: 1) images correctly classified by the model or 2) images misclassified by the model.

We trained a separate SVM for each character, since each character has a different entropy profile and different number of samples. We also used an 80/20 train/test split for each character. Additionally, due to the massive imbalance in the number of correctly and incorrectly classified characters, we balanced these data by choosing an equal number of correctly and incorrectly classified samples. Figure 12 depicts the results of this experiment, providing precision and recall statistics for both classes for all models and characters. It should be pointed out that it is unnecessary to display the precision *and* recall for both classes of a two-class classification problem since there is a dependency relation between these statistics. However, we chose to display both for ease in interpreting the results. For the CXE-ResNet, precision and recall values for both classes and all characters (other than $\sigma$) were between 0.486 and 1.000. For the KLD-ResNet, they were between 0.523 and 0.937. For the KNN, they were between 0.636 and 0.975.

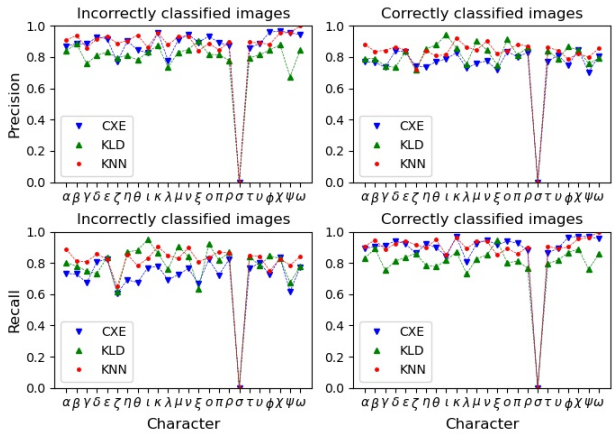


Figure 12: Per character precision and recall of the SVM model which predicts model classification accuracy based on Softmax entropy.

It should be noted that these SVMs tend to be biased toward predicting that the ResNets and ensemble models will correctly classify images. When examining their confusion matrices, there was a consistently larger number of false positives (where the SVM incorrectly predicted that the ResNets or ensemble models' classification would be correct) than false negatives. Also, it can be seen from Figure 12 that there is a consistent trend across the different characters where both the recall of the incorrectly classified images (bottom left) and the precision of the correctly classified images (top right) are lower than their converse values.

## 4 Conclusions

We have shown that by performing stacked generalization with a pair of well-performing ResNet models we are able to increase our prediction accuracy to a level beyond that achieved by either of the original models. We have also shown that the Shannon entropy of our models' output distributions is both an excellent measure of classification uncertainty and a predictor of classification accuracy for our data, with misclassified images tending to have a higher entropy than those correctly classified. Although our techniques have only been applied to a single crowdsourced dataset (AL-ALL), we believe it may be a useful approach for other datasets to reduce the effects of ground-truth uncertainty in labeling.

Future work may focus on the removal of ambiguous and mislabeled images from the AL-ALL dataset. There are a significant number of images in the dataset which have both a high HSM entropy and a high annotation count. Images such as these are often unintelligible, so removing them from the dataset may be beneficial when training models. Also, there are many images which are clearly intelligible but have been misclassified by the human annotators. These may be identified by using the ensemble model's results as the ground-truth which the human annotators must predict.

Images where the humans disagree with the model when the model is clearly correct (determined by visual inspection of the image) can either be corrected or removed.

## References


Bowman, A. K.; Coles, R.; Gonis, N.; Obbink, D.; and Parsons, P. J. 2007. *Oxyrhynchus: a city and its texts*. Graeco-Roman Memoirs, v. 93. London: Published for the Arts and Humanities Research Council by the Egypt Exploration Society.

C. E. Shannon, W. W. 1949. *The Mathematical Theory of Communication*. University of Illinois Press.

Deng, L. 2012. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6): 141–142.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. Adaptive computation and machine learning. MIT Press. ISBN 9780262035613.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Morin, M.; and Willetts, M. 2020. Non-Determinism in TensorFlow ResNets.

S. Kullback, R. A. L. 1951. On information and sufficiency. *Annals of Mathematical Statistics*, 22: 79–86.

Sorokin, A.; and Forsyth, D. 2008. Utility data annotation with Amazon Mechanical Turk. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 1–8.

Swindall, M. I.; Croisdale, G.; Hunter, C. C.; Keener, B.; Williams, A. C.; Brusuelas, J. H.; Krevans, N.; Sellew, M.; Fortson, L.; and Wallin, J. F. 2021. Exploring Learning Approaches for Ancient Greek Character Recognition with Citizen Science Data. In *2021 17th International Conference on eScience (eScience)*, 128–137. IEEE.

Swindall, M. I.; Player, T.; Keener, B.; Williams, A. C.; Brusuelas, J. H.; Nicolardi, F.; D'Angelo, M.; Vergara, C.; McOsker, M.; and Wallin, J. F. 2022. Dataset Augmentation in Papyrology with Generative Models: A Study of Synthetic Ancient Greek Character Images. In *The 31st International Joint Conference on Artificial Intelligence*. IJCAI-ECAI.

Williams, A. C.; Wallin, J. F.; Yu, H.; Perale, M.; Carroll, H. D.; Lamblin, A.-F.; Fortson, L.; Obbink, D.; Lintott, C. J.; and Brusuelas, J. H. 2014. A computational pipeline for crowdsourced transcriptions of Ancient Greek papyrus fragments. In *2014 IEEE International Conference on Big Data (Big Data)*, 100–105. IEEE.

Wolpert, D. H. 1992. Stacked generalization. *Neural Networks*, 5(2): 241–259.