# Accelerating Active Learning Image Labeling Through Bulk Shift Recommendations

Anonymous Authors

*Abstract*—**Nowadays, the inexpensive memory space promotes an accelerating growth of stored image data. To exploit the data using supervised Machine or Deep Learning, it needs to be labeled. Manually labeling the vast amount of data is time-consuming and expensive, especially if human experts with specific domain knowledge are indispensable. *Active learning* addresses this shortcoming by querying the user the labels of the most informative images first. One way to obtain the 'informativeness' is by using *uncertainty sampling* as a query strategy, where the system queries those images it is most uncertain about how to classify. In this paper, we present a web-based active learning framework that helps to accelerate the labeling process. After manually labeling some images, the user gets recommendations of further candidates that could potentially be labeled equally (bulk image folder shift). We aim to explore the most efficient 'uncertainty' measure to improve the quality of the recommendations such that all images are sorted with a minimum number of user interactions (clicks). We conducted experiments using a manually labeled reference dataset to evaluate different combinations of classifiers and uncertainty measures. The results clearly show the effectiveness of an uncertainty sampling with bulk image shift recommendations (our novel method), which can reduce the number of required clicks to only around 20% compared to manual labeling.**

*Index Terms*—**Active Learning, Computer Vision, Incremental Classification and Clustering, Image Classification, Image Labeling, Image Recognition**

## I. Introduction

Throughout the past decades, the rapid growth of image data was accelerated by an increased use of mobile phone cameras and the availability of cheap storage. Unfortunately, most of the generated images are unlabeled. To structure and classify images using supervised machine learning algorithms such as logistic regression or a convolutional neural network, a large number of labeled images is necessary. 'Manually' labeling all images would be an uneconomical and time-consuming process. There is a large volume of published studies describing the role of active learning in image classification approaches. The majority of prior research has shown the effectiveness of various sampling methods by applying appropriate active learning strategies, such as uncertainty sampling with the *entropy measure* [1] and refined methods taking retraining into account [2]. In uncertainty sampling, the system queries those instances next, about which it is most uncertain how to label. This approach has been improved by Papp and Szucs introducing a new query strategy for active learning using uncertainty sampling, the *entropy measure* with a *penalty metric* [1]. It has been shown that this new approach called *balanced active learning technique*, achieves the highest accuracy compared to using random sampling and uncertainty sampling. Similar results are achieved by Yang and Loog, demonstrating a new retraining-based active learner that integrates uncertainty information, showing an improvement of the accuracy in retraining-based models [2]. However, these approaches only compare the accuracy of different query strategies. They do not focus on user interactions.

In this paper, we present an active learning image labeling system that accelerates the sorting process through clicks-saving bulk image shift recommendations. This is a novel approach, which has not been explored so far to the best of our knowledge. we are aiming to answer the following research questions: 1) Which classification algorithm performs best while maximizing *time savings*? 2) Which uncertainty measure is most efficient in ranking the image candidates, such that the sorting requires a *minimum of user interactions*? 3) Which combination of classifier and uncertainty measure is fast enough to be used in an interactive setting? We employ a 'manually' labeled reference dataset to compare the performance (saved clicks, rounds, etc.) and runtimes of different models (classifiers) and samplings (uncertainties).

## II. Motivation

In our company, we have several mobile applications using machine learning approaches such as recommender systems or image recognition. Recently, we developed a cloud-based e-commerce shop, which allows managing product articles from diverse manufacturers. Additionally, the system has a built-in product recognition tool that can detect similar products based on their product image. However, to accomplish this, we need labeled data to classify images. With a large amount of image data, 'manually' labeling images would be time-consuming and inefficient. Therefore, we are exploring different active learning methods that minimize the number of user interactions needed for image labeling. If all images are labeled, they can be used to train an image classifier, e.g., a deep convolutional neural network [3].

## III. Background

Image classification belongs to the field of Computer Vision, which aims at recognizing visual patterns and predicting the corresponding class of an image [4]. Deep neural networks (DNNs) are considered the state-of-the-art algorithms used to solve image classification problems. In 2010, the ImageNet Large Scale Visual Recognition Competition (ILSVRC) started, aiming to measure the progress of image recognition comparing different DNNs [5]. In supervised machine learning [6] we need labeled training instances so that we can perform classification or regression algorithms [7]. On the contrary, in unsupervised learning approaches, all training instances are unlabeled. However, when addressing classification problems, having a large amount of unlabeled instances requires 'manual' annotation of the training data, which is time-consuming and thus cost-expensive [8].

*Active learning* is a subfield of machine learning that enables the selection of the most valuable instances, which have to be labeled first instead of randomly selecting a set of unlabeled instances [9]. The active learning cycle starts by 'manually' labeling a few randomly chosen instances of the unlabeled pool. Thereby, we create a machine learning model based on the labeled training set. Depending on the query strategy on the unlabeled pool, the model asks the human annotator for the label, which provides the most relevant information. The labeled instance is then added to the training set. This way, the active learning model achieves high efficiency by 'manually' labeling as few instances as possible. [9]. There are different active learning scenarios. The most common are *stream-based selective sampling* [10] and *pool-based sampling* [11]. Here we focus on the latter. Applying *pool-based sampling* methods is efficient to query a large collection of unlabeled data at once. The active learning process starts with labeling a few examples to train a classifier. A subset of unlabeled instances is then extracted from the data pool according to a defined informativeness measure for the classifier. Subsequently, all unlabeled images in the subset are ranked by their informativeness, presented to and labeled by a human annotator [12]. One of the most used active learning methods to define the informativeness is to use an uncertainty measure, where a learner selects the instances that the model is least certain how to label [13].
Querying the least certain label can help the model to improve its accuracy efficiently [14]. Such an approach is adequate for models with probabilistic outcomes. There are several possibilities to specify uncertainty measures [9]. A common sampling method [1] is to query the instances, for which the model has **least confidence** in their most likely label. This can be written as [9]

$$x^*_{LC} = \text{argmax}_x \;\; 1 - P_\theta(\hat{y}|x), \tag{1}$$

where $P_\theta(\hat{y}|x)$ is the probability of the most probable label $\hat{y}$ of instance $x$ predicted by the model $\theta$. The larger $x^*_{LC}$, the more likely it is that the model labels the instance incorrectly. The least confident score only focuses on the most probable class. The remaining classes are only indirectly taken into account via the condition that the probabilities need to sum up to one. A first attempt to directly include more information about the label distribution is **margin sampling** [15]. **Margin sampling** determines the probability difference between the first and second most probable classes $\hat{y_1}$ and $\hat{y_2}$ to find the instance $x$, where

$$x^*_M = \text{argmin}_x \;\; P_\theta(\hat{y_1}|x) - P_\theta(\hat{y_2}|x) \tag{2}$$

Employed in pool-based sampling, the model selects the instances with small margins, which are most difficult to distinguish by the classifier. Another commonly used uncertainty measure that takes into account the entire label distribution [16] is the **Shannon entropy**:

$$x^*_H = \text{argmax}_x \;\; -\sum_i P_\theta(y_i|x) \;\; \log P_\theta(y_i|x), \tag{3}$$

where $y_i$ covers all possible labels with their conditional probability $P_\theta(y_i|x)$. The entropy quantifies the average amount of information needed to find the true class label.

Active Learning has been employed for image classification and labeling already for some time. In 2013, Li et al. present an adaptive framework that combines an information density measure with uncertainty sampling [17]. In 2016, Ye et al. introduce a model based on a bag-of-visual-words (BoVW) uncertainty measure and structural similarity (SSIM) of the images [18]. In 2017, Wu et al. address the problem of sample noise and missing label correlation exploitation with an adaptive low-rank multi-label active learning algorithm, (LRMAL) [19]. In 2019, Wang et al. propose a novel method that combines uncertainty, diversity, and density via sparse modeling in the sample selection to address the problem of redundant information across samples [20]. Most recently in 2021, Li et al. present an active deep learning framework for image classification [21]. However, to the best of our knowledge, there has not been an approach so far that uses *bulk image shifting* to accelerate the active learning process.

## IV. Methodology

This section presents the architecture and workflow of our implementation of an active learning system for image sorting, along with a description of the employed reference dataset. For our experiments, we use around 3000 product images, which were 'manually' sorted into 33 categories such as suitcases, chairs, trash cans, or hammers. We built a web application that provides a user-friendly active learning image sorting system. It allows users to label all images of a dataset with as few interactions as possible. We employ the open-source framework *Angular* [22] for our front-end to handle the client interactions, and the python-based micro-framework *Flask* [23] as middleware. Furthermore, we use the neural-network library *Keras* [24] to generate image-encodings using a pre-trained CNN. Lastly, we employ the machine learning library *Scikit-learn* [25] for the image clustering and classification. The communication is done via a *REST API*, provided by *Flask*.

The starting point of our active learning system is the upload of a folder containing unsorted images. The images are then converted into feature vectors of size 2048 generated from the second but last layer of a VGG16 network pretrained on the ImageNET dataset, and provided in Keras [24]. Next, a presorting using *k-means* clustering allows to label larger batches of images (folders) in one click. Users only have to go over the folders created by the clustering and name them after looking at the images in the folder and trying to understand what they have in common. After this (nearly) fully unsupervised image pre-clustering, the system allows for semi-supervised classification using active learning. In several sampling rounds (in the first, the labeled clusters are taken as classes, see Figure 1), a specified classification algorithm is executed. The system then queries the user the labels of 250 images it is most uncertain about, based on a selected sampling measure.

Each image candidate has four attributes:

- **Actual label** - the correct label (as in the 'manually' labeled reference dataset) used for testing only.
- **Current label** - the label of the folder, where the image is currently located.
- **Recommended labels** - the first ten labels, sorted by the descending uncertainty of the classifier.
- **Remaining labels** - all labels except the recommended labels.

The next step is to label the images correctly by clicking on one of the recommended labels or the remaining labels. If we click on a label in the remaining labels, the image is moved to the selected folder. By choosing a label from the recommended labels, the system proposes to move further images if a) all images have the same current label (same source folder), and b) all images have the same recommended label at the position of the clicked label (same destination folder). For example, if the user accepted a recommendation to shift an image from 'label 1' to 'label 3' by clicking 'label 3' being the second-ranked recommendation, the system retrieves all images that are currently labeled as 'label 1' and have 'label 3' also as a second-ranked recommendation. In Figure 2 we see an example of a recommendation for a bulk image move after clicking on the label **'Schrank'**[1] of an image. All recommended images have the same current label **'Regal'** because we selected an image whose label is **'Regal'**, and the position of the recommended label **'Schrank'** is identical in each image. Thus, more images can be moved using fewer user interactions ( [1]). The completeness defines the percentage of the images that are labeled according to the 'manual' reference sorting. Each shift of an image will potentially lead to a recommendation to move more images, which can, if accepted, increase the completeness significantly. After sorting the first round, we repeat the whole process until we have achieved full completeness of approximately 100%[2]. As a measurement of how well the classifier performs, we track the number of images the user has seen, shifted, and the number of clicks the user made during sorting. To accelerate the evaluation, we have developed an automation script that simulates a user sorting images correctly according to their actual (reference sorting) label.

## V. Evaluation

In this section, we will describe our experimental approach to find the optimal active learning strategy, evaluating a combination of several classifiers and different uncertainty measures. Subsequently, we present the results of our evaluation using a variety of test-metrics to assess the contributions of the individual test-parameter. The *move more statistics* describe the effectiveness of image candidate recommendations, which could belong to the same class folder as the selected candidate. We use the purity of a recommendation as a measure for its quality, defined as purity $= \frac{n_c}{n_s}$, where $n_c$ is the number of correctly recommended images and $n_t$ is the total number of recommended images. The higher the purity of each bulk shift recommendation, the fewer clicks the user has to make during the sorting process. In each round in our pool-based sampling, the systems recommends 250 uncertain image candidates, returned after the execution of a specific classifier. We applied a *logistic regression*, *multi layer perceptron*, and *decision tree* classifier to execute the classification tasks. For multi layer perceptron, we set *ReLU* as activation function and utilize *Adam* as solver, which is a gradient-based optimizer performing well on large datasets. The decision tree classifier uses the *gini* criterion to measure the quality of a split. For each classifier, we applied *margin sampling*, *shannon entropy*, and *least confidence* as respective uncertainty measures. To demonstrate our approach, we plot in Figure 3 the completeness (fraction of labeled instances in the dataset) vs. the number of clicks normalized by the total number of images. Benefiting from the pre-clustering, we start with a completeness of $0.668$. This means that 67% of the images were already grouped correctly by the unsupervised clustering. The user only has to label the cluster folders (which we do not count as click). Further, the positive effect of bulk moves (counted as a single click) are visible in Figure 3 as vertical jumps. As a measure for the performance, one could use the fraction of clicks needed to reach a completeness of 100%. However, this would not discriminate how this point is reached. Inspired by the ROC-analysis, we use the area under the learning curve AUC as a more complete measure for the performance. A perfect curve reaching 100% completeness with one click would have an AUC of 1.0. In contrast, an almost perfect curve reaching 99% with one click but needing many more clicks afterwards to finally reach 100% would also have an AUC close to 1. The AUC values of the different classifiers and sampling methods are shown in Table I.

## VI. Results

This paper aimed to determine the best performing active learning approach, combining a classifier with an uncertainty measure. Table I contains the values for the area under the active learning curves as a measure for the click-savings of the process. Data, code, result tables, and a demovideo are publicly available at https://github.com/AnonymousCSResearcher/BulkImageLabelShifting. The README explains how the results can be reproduced. Our results reveal that the logistic regression classifier with either shannon entropy or least confidence as uncertainty measure are the most efficient combinations in terms of AUC (0.963) and also more efficient than the multi layer perceptron and the decision tree, see Table I. Logistic regression with shannon entropy or least

[1] In our example dataset, we used German labels.

[2] We set the cutoff threshold to 99.9% since the user could accidentally sort some images differing from the 'manually' labeled reference set.

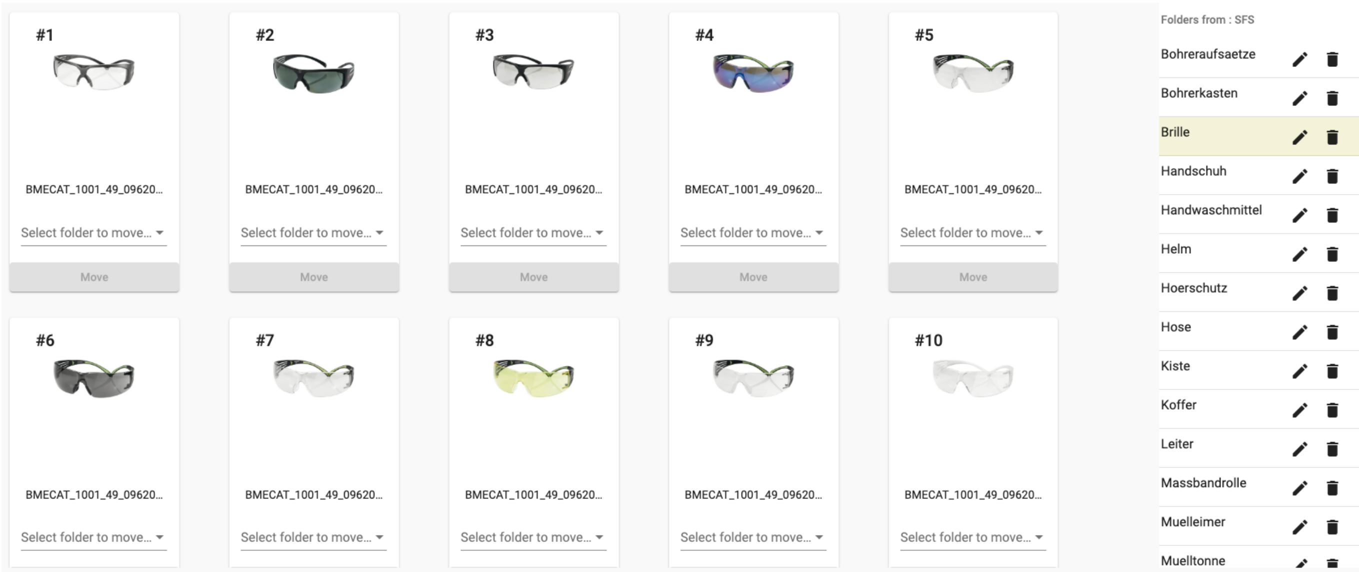


Fig. 1. After the presorting using *k-means* clustering, the user can label the image folders as a starting point for the active learning image shift recommendation process.

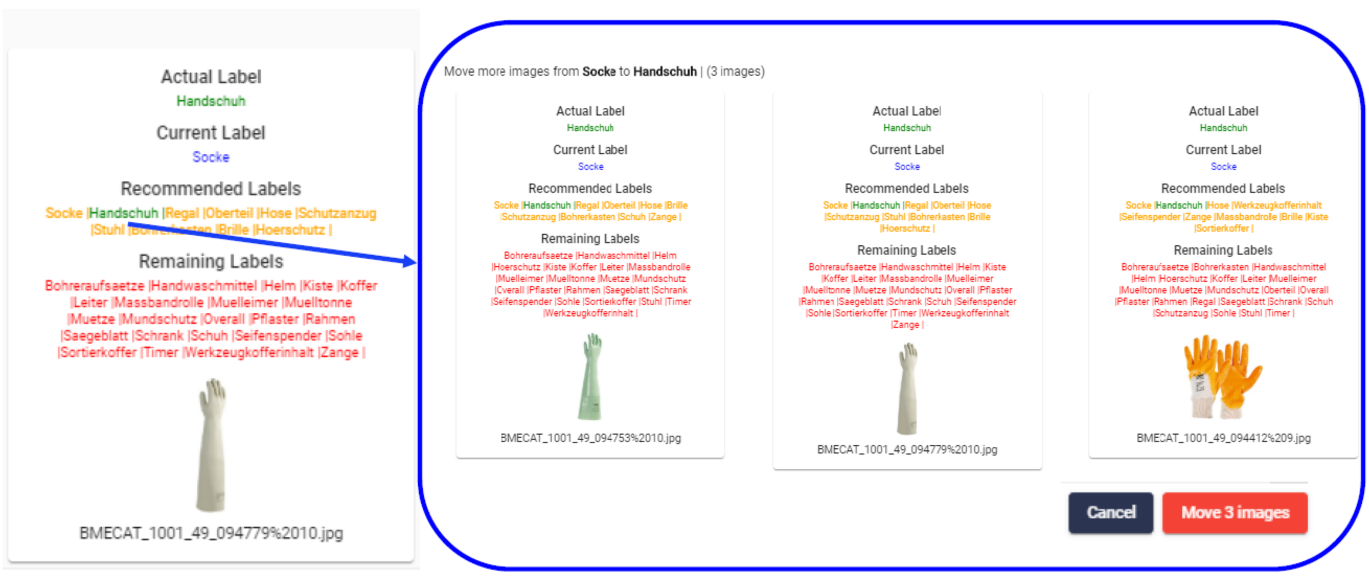


Fig. 2. Recommendation of further images that additionally could be moved based on the selected source and destination (here: move from the (German) label **'Socke'** to **'Handschuh'**).

confidence are also the best combinations for reducing the number of clicks until reaching full completeness. Labeling approximately 22% of the image dataset is already sufficient to have all images sorted. Comparing the runtimes of the classifiers (Table II), one can see that the decision tree is by far the fastest algorithm on average, only taking half as long as the other classification models. Multi layer perceptron requires 61% more runtime compared to decision tree. Furthermore, margin sampling provides the highest average number of recommended bulk moves and a maximum mean purity of nearly 92%, whereas least confidence performs worst in both test-metrics.

## VII. Discussion

Our findings suggest that applying Shannon entropy as uncertainty measure significantly reduces the number of user interactions (clicks) required to sort the image dataset but turns out to have the highest runtime in the execution. The fact that the Shannon entropy sampling outperformed the other sampling methods, is probably because of the efficient calculation of the informativeness by considering each image candidate in the probability distribution, as demonstrated by Papp and Szucs [1]. Surprisingly, the least confidence approach competes against Shannon entropy in the final accuracy as

TABLE I

AREA AUC UNDER THE ACTIVE LEARNING CURVE, WHICH CORRELATES WITH THE CLICK-SAVINGS OF THE PROCESS. THE GREEN/RED CELL COLOR INDICATES MINIMUM/MAXIMUM VALUES.

| **Classifier / Uncertainty Measure** | *Shannon Entropy* | *Margin Sampling* | *Least Confidence* | **Mean** |
|---|---|---|---|---|
| *Logistic Regression* | 0.963 | 0.934 | 0.963 | 0.953 |
| *Decision Tree* | 0.937 | 0.937 | 0.892 | 0.922 |
| *Multi Layer Perceptron* | 0.955 | 0.934 | 0.946 | 0.945 |
| **Mean** | 0.951 | 0.935 | 0.934 | |

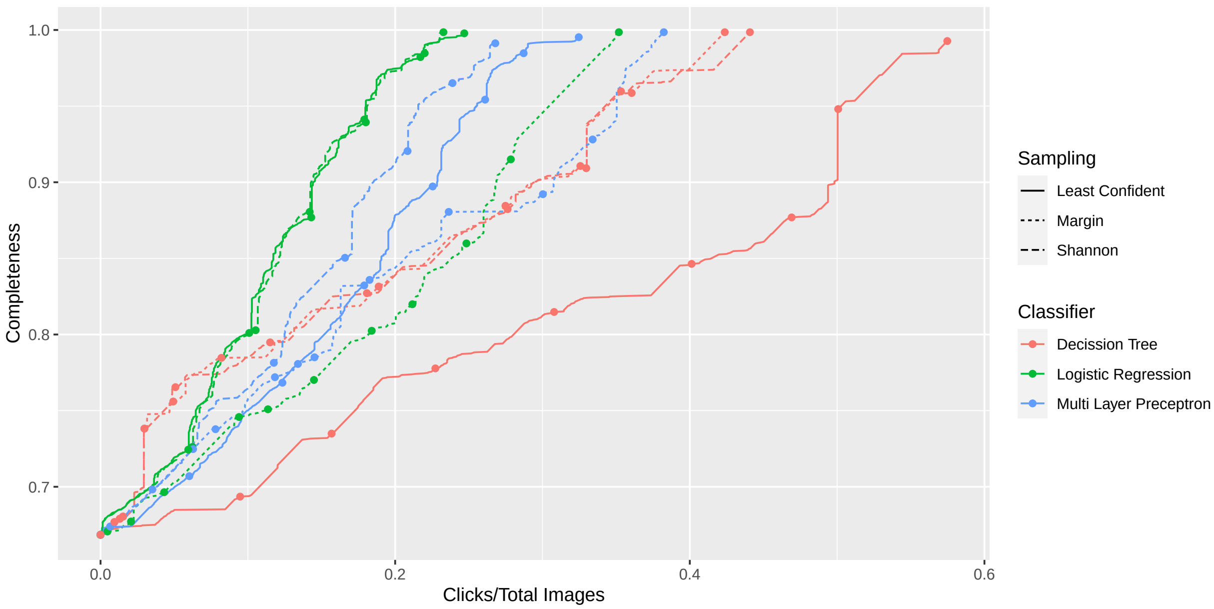


Fig. 3. Evaluation of different classifier and sampling methods. Due to the pre-clustering, the completeness already starts at 0.669. The individual sampling rounds are indicated with dots. Sudden vertical jumps in the completeness, correspond to successful bulk image moves.

TABLE II

RUNTIME IN SECONDS (4 GCP VCPUS, 16 GB RAM) REQUIRED FOR SIMULATING THE ACTIVE LEARNING PROCESS UNTIL THE CUTOFF THRESHOLD OF 99.9% COMPLETENESS. THE GREEN/RED CELL COLOR INDICATES MINIMUM/MAXIMUM VALUES.

| **Classifier / Uncertainty Measure** | *Margin Sampling* | *Least Confidence* | *Shannon Entropy* | **Mean** |
|---|---|---|---|---|
| *Logistic Regression* | 1160.15 | 772.91 | 1131.84 | 1021.63 |
| *Multi Layer Perceptron* | 1631.65 | 1097.40 | 1719.03 | 1482.69 |
| *Decision Tree* | 614.03 | 501.73 | 590.78 | 568.85 |
| **Mean** | 1135.28 | 790.68 | 1147.21 | |

well as in the percentage of saved clicks. Evaluating the performance of the classifiers, we see that logistic regression performs best, followed by the multi layer perceptron, and the decision tree. We speculate that this is because decision trees tend to produce uncalibrated probabilities [26] and so harm the sampling methods requiring proper probability estimates. Concerning the runtimes of the classifiers, decision tree is by far the fastest one but yet requires a large amount of image data to be 'manually' labeled (percentage of clicks is 60%, large compared to the settings mean of 38%). This could be because the depth of the tree is not limited, which may lead to overfitting the model. The logistic regression shows contrary results by having a three times higher runtime but requiring fewer images to be clicked until full completeness. These two test metrics, the runtime of the classifier execution and the percentage of image labeling clicks, are a trade-off interpreting the results. A major problem in our experimental approach was a high runtime when executing various classification algorithms, which is most likely caused by the large size of the image vectors (dimension 2,048). This can slow down the execution process.

## VIII. Conclusion and Outlook

In this paper, we presented our active learning recommender system for image sorting to label for supervised classification. The novelty of our approach is the acceleration through bulk image shifts, which require only a single click (vertical jumps in the completeness vs. clicks curve). We examined the efficiency of making use of several different uncertainty-based sampling strategies and classifiers. Based on our experimental evaluation, where we employed a dataset containing around 3000 images, it can be concluded that combining logistic regression with shannon entropy or least confidence is most efficient in helping the user to sort the images quickly. In this case, clicking only 22% of the images in the dataset results in maximum completeness. In general, our active learning system demonstrates that it is possible to significantly accelerate image labeling. By iteratively querying the user the labels of those images with the highest classification uncertainty, the image sorting quickly achieves full completeness.

In our future work, we will examine whether further sampling methods can increase click-savings and reduce the number of image labeling clicks. Furthermore, we will test other classifiers, such as *support vector machine*, *calibrated trees* or *k-nearest neighbors*, etc. Moreover, we plan to explore the stream-based learning approach, where the active learning algorithm determines 'on-the-fly' whether to query the label of a specific instance or drop it based on an assessment of its informativeness. So far, we have only applied pool-based sampling. Furthermore, we intend to deploy the active learning system in the cloud to provide a time-saving, economical image sorting system for our employees, as our e-commerce shop requires labeled product images to perform *product image recognition*. Lastly, we aim to tackle the challenge of not only classifying a broad product category but recognize specific products, e.g., using one-shot learning [27] or a multistage process containing product classification and ranking.

## References


[1] Dávid Papp and Gábor Szücs. Balanced active learning method for image classification. *Acta Cybern.*, 23:645–658, 2017.

[2] Y. Yang and M. Loog. Active learning using uncertainty information. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2646–2651, Dec 2016.

[3] Jeff Heaton. Ian goodfellow, yoshua bengio, and aaron courville: Deep learning - the MIT press, 2016, 800 pp, ISBN: 0262035618. *Genetic Programming and Evolvable Machines*, 19(1-2):305–307, 2018.

[4] Nilanjan Dey, Girish Mishra, Jajnyaseni Kar, Sayan Chakraborty, and Siddhartha Nath. A survey of image classification methods and techniques. 07 2014.

[5] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[6] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, 2014.

[7] S B Kotsiantis. Supervised machine learning: A review of classification techniques. *Informatica*, 31(3):249–268, 2007.

[8] Dagoberto Cruz-Sandoval, Jessica Beltrán-Márquez, Matias Garcia-Constantino, Luis A. González-Jasso, Jesús Favela, Irvin Hussein López-Nava, Ian Cleland, Andrew Ennis, Netzahualcóyotl Hernández, Joseph Rafferty, Jonathan Synnott, and Chris D. Nugent. Semi-automated data labeling for activity recognition in pervasive healthcare. *Sensors*, 19(14):3035, 2019.

[9] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[10] Andreas Kvistad, Massimiliano Ruocco, Eliezer de Souza da Silva, and Erlend Aune. Augmented memory networks for streaming-based active one-shot learning. *CoRR*, abs/1909.01757, 2019.

[11] Andrew McCallum and Kamal Nigam. Employing em and pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 350–358, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[12] Sudheendra Vijayanarasimhan and Kristen Grauman. What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 2262–2269. IEEE Computer Society, 2009.

[13] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 3–12, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[14] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1070–1079. ACL, 2008.

[15] Jin Zhou and Shiliang Sun. Improved margin sampling for active learning. In Shutao Li, Chenglin Liu, and Yaonan Wang, editors, *Pattern Recognition - 6th Chinese Conference, CCPR 2014, Changsha, China, November 17-19, 2014. Proceedings, Part I*, volume 483 of *Communications in Computer and Information Science*, pages 120–129. Springer, 2014.

[16] Catuscia Palamidessi and Marco Romanelli. Feature selection in machine learning: Rényi min-entropy vs shannon entropy. *CoRR*, abs/2001.09654, 2020.

[17] Xin Li and Yuhong Guo. Adaptive active learning for image classification. In *CVPR*, pages 859–866. IEEE Computer Society, 2013.

[18] Zhipeng Ye, Peng Liu, Jiafeng Liu, Xianglong Tang, and Wei Zhao. Practice makes perfect: An adaptive active learning framework for image classification. *Neurocomputing*, 196:95–106, 2016.

[19] Jian Wu, Anqian Guo, Victor S. Sheng, Pengpeng Zhao, Zhiming Cui, and Hua Li. Adaptive low-rank multi-label active learning for image classification. In *ACM Multimedia*, pages 1336–1344. ACM, 2017.

[20] Gaoang Wang, Jenq-Neng Hwang, Craig S. Rose, and Farron Wallace. Uncertainty-based active learning via sparse modeling for image classification. *IEEE Trans. Image Process.*, 28(1):316–329, 2019.

[21] Guoqiang Li and Ning Gong. Substep active deep learning framework for image classification. *Pattern Anal. Appl.*, 24(1):23–34, 2021.

[22] Angular - web application framework. https://angular.io/. Accessed: 2019-08-23.

[23] Flask - python-based micro web-framework. https://palletsprojects.com/p/flask/. Accessed: 2019-08-23.

[24] François Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[25] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011.

[26] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632, 2005.

[27] Weidong Geng, Feilin Han, Jiangke Lin, Liuyi Zhu, Jieming Bai, Suzhen Wang, Lin He, Qiang Xiao, and Zhangjiong Lai. Fine-grained grocery product recognition by one-shot learning. In Susanne Boll, Kyoung Mu Lee, Jiebo Luo, Wenwu Zhu, Hyeran Byun, Chang Wen Chen, Rainer Lienhart, and Tao Mei, editors, *2018 ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea, October 22-26, 2018*, pages 1706–1714. ACM, 2018.