# Wally – Crowd powered image matching on tablets

Deepak Pai
Adobe Systems India Pvt Ltd
Bangalore
India
91-80-41939984
deepak.pai@adobe.com

James Davis
UC Santa Cruz
California
USA
1-650-799-2574
davis@cs.ucsc.edu

## ABSTRACT
In this paper we propose a crowd sourced approach for solving large scale object retrieval. We have built a tablet application which displays a query image and a database image. The crowd provides their input to indicate, if there is a match between the query and database image or not. We test our application on a crowd of low-income individuals. We observe that our target crowd had a very high accuracy on the considered dataset. We observe significant improvement as compared to vision based image matching algorithms available in prior-art. We also observe that with simplistic interfaces, even low literacy and low income people could participate in the crowdsourcing tasks. This provides them a significant income opportunity. We have validated our claims on two publicly available University of Kentucky datasets and ORL Face recognition dataset.

## Categories and Subject Descriptors
H.4.0 **[Information System Applications]**: General;
H5.m. **Information interfaces and presentation (e.g., HCI):** Miscellaneous.

## General Terms
Experimentation, Human Factors.

## Keywords
Crowdsourcing, Human Computation, Mechanical Turk, Micro-tasks, Computer Vision, Image Search, Mobile Crowdsourcing.

## 1. INTRODUCTION
Image matching and retrieval from databases is an important component of many large software systems. For example, consumer photo editing software has features for finding similar images in a photo collection, and biometric security systems require face recognition from large databases. Even nearly identical images can sometimes be hard for automated algorithms to match, and the problem is compounded by lighting variation, change in viewpoint, orientation, scale, expression, aging, and variations in background.

While it is difficult for automated algorithms to robustly identify two matching images or faces, humans can often achieve this task easily. Unfortunately it would be impossible for humans to scan millions or billions of images in a database, so the task cannot simply be transferred completely to humans.

This paper investigates a very simple hybrid algorithm that first uses an automated algorithm to sort matches, and then uses human labor to prune false positive matches from the list. Although much more sophisticated hybrid algorithms exist, even this very simple method has much better accuracy than the pure automated algorithm, and thus it is worth characterizing. The price of improved accuracy is that human labor needs to be obtained and paid.

Paid crowdsourcing services such as Amazon Mechanical Turk provide a platform for algorithms to obtain human labor, and humans to engage in micro-task. However these platforms are often too complex for low-income low-literate potential workers. Since the ultimate price of a hybrid algorithm is correlated with the efficiency and pay rate of human labor, we suspect that low income workers will ultimately produce the most "efficient" algorithms.

In order to investigate the potential of low literate workers to perform the work necessary for hybrid image retrieval algorithms, we built a custom "work" application and tested worker accuracy and efficiency directly with low paid office workers in Bangalore. Current generations of mobile phones and tablets are capable of displaying reasonably good quality images. They have touch based inputs and can be easily operated by our target crowd. We created a simple interface, in which workers only need to indicate if query and target images match or not. As soon as input is provided, another pair of query and target images are shown, allowing workers to solve as many queries as possible efficiently. Our mobile application with only visual interfaces, serves as a prototype of how low literate workers may eventually be brought into the micro-task workforce. Our interest in this study, is obtaining demographics, efficiency, and accuracy estimates for what we believe may eventually be a different category of labor pool than currently exists.

The first contribution of this paper is an analysis of a simple hybrid image matching algorithm, showing that it outperforms a purely CPU based algorithm. The second contribution is showing that low literacy workers can effectively perform the kinds of tasks necessary for hybrid image matching algorithms.

The remainder of the paper is organization as follows, Section II discusses related work. Section III describes our application and target crowd. Section IV provides experimental setup and comparative results. Finally, conclusion and future work are presented at the end.

## 2. Related Works
Computer algorithms for image matching and face recognition from large scale databases are active areas of research [22, 23, 24, 25, 26]. Recently SIFT based approaches such as the work by Lin and Brandt have proved to be very effective in object retrieval from large scale databases [1] [2]. Similarly Zhang et al. show good results in the area of face recognition [3]. This paper concentrates on improving the results of automated algorithms by using human computation to detect false positives.

A variety of platforms for paid crowdsourcing have emerged, including Amazon Mechanical Turk, CrowdFlower, CloudFactory, MobileWorks, and Samasource. Researchers are treating human computation as a new computation platform and building programming tools to make deploying new applications easier [4, 11, 13, 14, 15]. Micro-work has shown promise in a wide variety of applications including image annotation [16], collecting training data [17], user studies [18], tools for the blind [19], document editing [12], and processing database queries [21, 20]. This paper explores whether hybrid human-machine computation algorithms can outperform machine only algorithms for image matching. It also explores the suitability of currently unreached low literacy workers as a potential worker pool for image matching tasks.

Paid micro-work potentially provides significant earning opportunities in developing countries like India [6]. However, user studies suggest that current interfaces require computer literacy and are beyond the reach of most low income workers [7]. Some researchers have targeted basic mobile platforms, since this population generally has access and is capable of using a mobile phone [8, 27, 28]. This paper explores an interface on tablet platforms, designed to provide more screen space than a phone, but still support low literacy workers.

## 3. Proposed Application

We developed a tablet based application to gather information about individual image matches.

**Figure 1: Wally application interface. The application is characterized by absence of textual interfaces.**

Figure 1 shows a screenshot of our application. The application does not have any textual interface. All elements are visual and easily understandable by the target crowd. The worker is presented with two images. The image on the left is the query image. The image on the right is the database image. The worker indicates if the images match or not. If there is a match, the tick mark is clicked. If images do not match, the button with a cross is clicked. Once input is provided, the next pair of images is displayed automatically. At any point the worker has an option to navigate to previous a pair of images and change their response. Once the worker has finished matching the stipulated set of image pairs the application stops and stores the results onto an xml file.

## 4. Experimental Setup and Results

### 4.1 Dataset

We considered two different datasets to evaluate our method. The first is the University of Kentucky (UKBench) dataset for image matching [9]. This consists of 2550 different images. Each image has four different variations under different conditions making a total of 10200 images. The variations are in the form of change in view point, lighting, scale, orientation etc. Sample images from the dataset are shown in Figure 2.

The second dataset is the ORL database comprised of faces of 40 different people [10]. Every face has 10 different images. The images vary in lighting, facial expressions (open / closed eyes, smiling / not smiling), facial details (glasses / no glasses) and time. In total this dataset has 400 images. Sample images from the dataset are shown in Figure 3.

**Figure 2: 4 different images from UKBench with corresponding matches.**

**Figure 3: 4 different images from ORL with corresponding matches.**

### 4.2 Experimentation Details

One of the primary intentions of our work is to investigate whether low-income and low literacy people can successfully participate in crowdsourcing tasks. Our subjects were chosen from professions such as security guard, office housekeeping staff, and catering staff. The user was briefed about the experiment and the application for a couple of minutes before using the application. There was no further guidance during the experimental work periods. The experiment was conducted with a Samsung galaxy tablet.

We experimented with ten different people in our target group. Each experiment had ten different query images. The human was asked to match each query image against ten other images from the dataset. These ten images were the top ten matches for the query image as returned by the automated algorithm from Lin and Brandt [2]. Thus, the workers labeled 80 random query images from UKBench, each against the top 10 images from the automated algorithm. In the case of the ORL face recognition dataset, we labeled 30 random query images, against the 10 top images. In total 800 image pairs from UKBench and 300 pairs from ORL were labeled.

**Accuracy of low computer literacy workers:** On the UKBench dataset, our target workers achieved a very high accuracy of 98.2% in identifying correct matches for a given query image. We separately quantified their accuracy as 99.2% at identifying incorrect matches. On the ORL dataset, correct matches to a query image were identified at a rate of 97.5%, while incorrect matches were located with accuracy of 98.5%.

The reported values are single respondent accuracy. It is common in human computation to sample multiple workers and aggregate information to obtain higher accuracies [29, 30]. Our goal is to investigate the possible value of hybrid algorithms, not to present the best possible method. Thus we do not aggregate results in any way, and simply rely on the accuracy rate of single workers in the analysis below.

## 4.3 Results

Our method is the simplest possible hybrid human-machine computation for image matching. We choose an existing automated method [2], let it rank the target images and select the top K as matches. Then, human workers attempt to prune the false positives from the results, using one worker response per image.

We investigate this new method in three ways: the impact on false positives presented to end users, comparing the ROC curve of the new algorithm to the existing algorithm, and the cost in dollars to obtain improved accuracy.

**False positives presented to end users:** One common use of image matching algorithms is to produce search results, such as might occur in a consumer photo editing application that searches your personal image archive for near matches to a specified image. Existing implementations might rank the database of images, and present the top K images on the screen. The usefulness of this interface depends strongly on the number of false positives presented. End users (as distinct from human workers who are part of computation algorithms) have no patience to look at wrong matches. Ideally only true positive matches would be presented.

We let the automated algorithm determine the top K matches, all of which would be presented on the screen. Our algorithm pruned this set to remove the false positives. Figure 4 compares the number of incorrect images displayed to the end users, using the machine algorithm and our hybrid algorithm on UKBench dataset. Since the workers sometimes make mistakes, a small number of false positives are still presented to the end user, but the number is dramatically reduced. Nearly all incorrect matches are removed.

The ORL Face dataset behaves similarly, as seen in Figure 5. Using a hybrid algorithm would allow an application to almost completely avoid showing incorrect matches to a user.

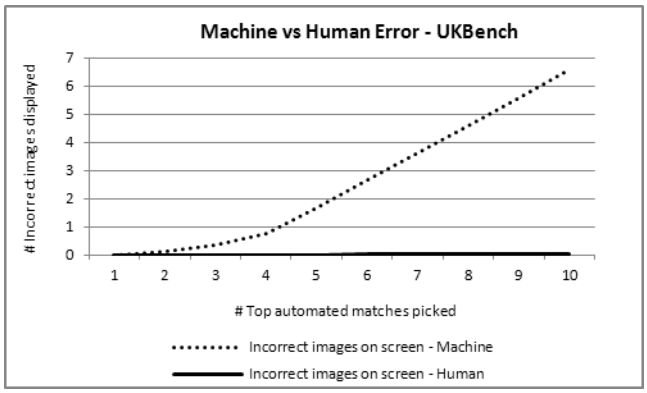


**Figure 4: Comparative error rates for the machine and human algorithm on UKBench.**

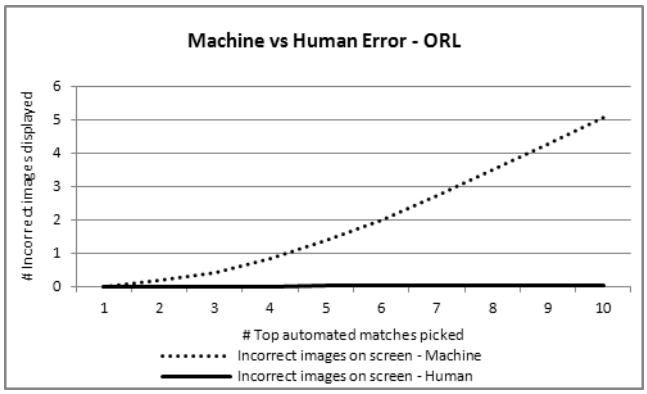


**Figure 5: Comparative error rates for the machine and human algorithm on ORL.**

**ROC Curves:** In any proposed algorithm we are interested in both the true positive rate and false positive rate. As we obtain human labels we generate both true and false positives.

A comparison of the ROC curve for each algorithm on the UKBench dataset is plotted in Figure 6. This is the dataset that was used when the machine algorithm was published, and it performs extremely well. Nevertheless, false positives can be pruned by using human judgments, and our hybrid algorithm has a better ROC curve. Note that we did not actually obtain human labels for the entire dataset. Instead we extrapolated using the measured accuracy of our workers. Figure 7 zooms in on top left corner of Figure 6. Human workers correctly prune nearly all false positives, but only correctly identify 98.2% of true positives. After this near vertical section of the curve, our hybrid algorithm does not have an advantage and flattens out.

A comparison of ROC curves for the ORL Face dataset is plotted in Figure 8. This dataset is more difficult for the machine algorithm, but the hybrid algorithm performs very well.

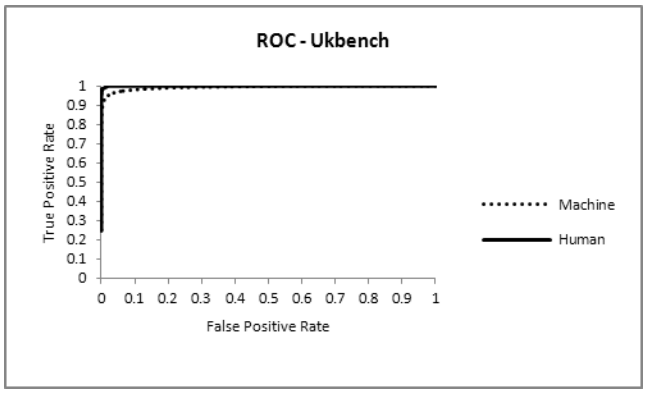


**Figure 6: ROC - Machine vs Human on UKBench.**

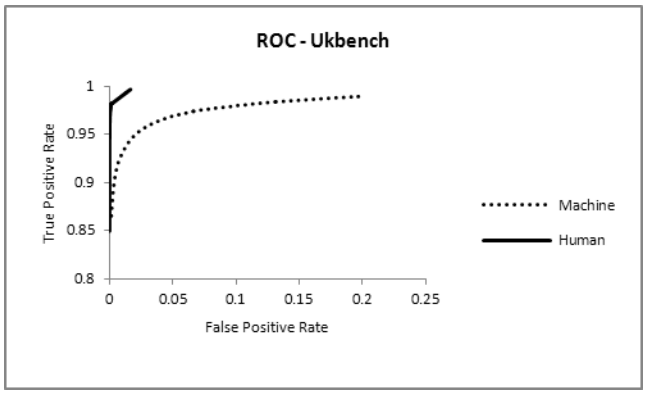


**Figure 7: ROC curve in Figure 6 zoomed in on top left corner.**

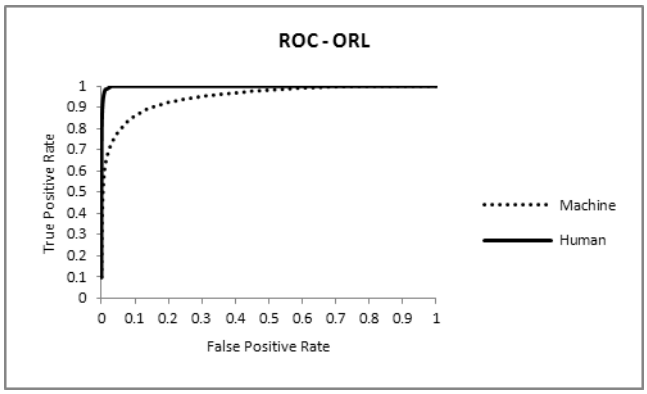


**Figure 8: ROC - Machine vs Human on ORL.**

**Cost of improved accuracy:** Our hybrid machine-human method is based on paid micro-task. Thus it is not free to make a query, and the cost depends on the pay rate of the underlying human computation. The people in our target population earn approximately $100 per month. In 10 minutes they were able to label about 100 images. Hence the cost of matching one query-test image pair is as low as ~0.07 cents. We paid the workers at this rate, proportional to the earning rates from their day jobs.

If the machine algorithm returned all the true positives in the top K matches, workers would only need to give judgments on K image pairs. However in some cases the machine algorithm ranks a true positive match very low, requiring many pairs to be checked before it can be found. Figure 9-10 show the amount that would need to be spent for human judgments to be obtained on a given percentage of all images with all matches. For example, for the hybrid algorithm to have found all of the matches for 90% of the images in the UKBench dataset, approximately 400 image pairs will need to be tested at a cost of approximately $0.26.

The value of producing high quality results will vary with application, and $0.26 might be either high or low relative to the domain. However we do not expect that this simple hybrid algorithm would be used to exhaustively search a very large dataset for all true-positives. Instead we think the value lies in pruning false-positives.

Many applications require a very low false positive rate, especially when results will be displayed to end-users. Workers can look at the top 50 matches from a machine algorithm for less than 5 cents, and prune nearly all false positives. Consider a face recognition deployment meant to locate criminals. An algorithm that returns 1 true positive for each 50 false positives is not likely to be usable. However this simple hybrid algorithm would change the ratio to 1-1 for a few cents per query.

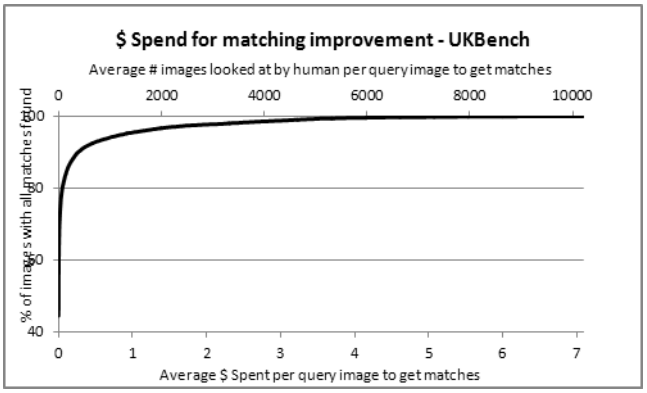


**Figure 9: $ Amount to be spent for improving UKBench.**

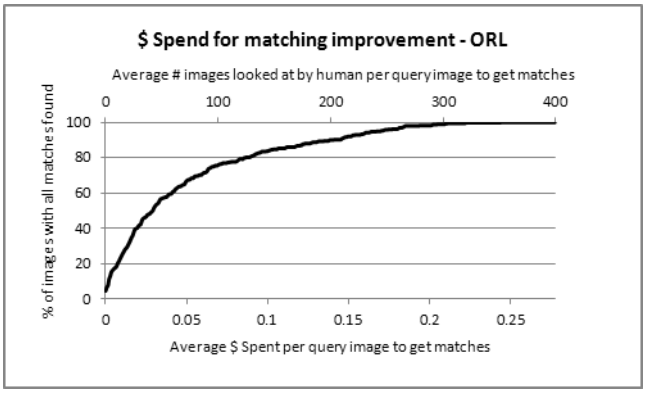


**Figure 10: $ Amount to be spent for improving ORL.**

## 5. Conclusion

We have investigated a simple hybrid human-machine algorithm for image matching. We find that the algorithm performs significantly better than the comparison machine-only algorithm.

We have also investigated whether potential workers with low computer literacy might be suitable for performing the image matching micro-tasks necessary to power our algorithm. We created a tablet application with minimal user interface, and tested efficiency and accuracy of workers from the target population. We conclude that they *can* accurately perform the task.

We have experimented with a small set of query images to prove the feasibility of our proposed approach. We believe that future

work should focus on more sophisticated hybrid algorithms, in terms of interaction between the human and machine components, as well as more fully considering the error rates of the humans by bringing in redundancy and agreement protocols.

## 6. REFERENCES


[1] D. Lowe, "Object recognition from local scale-invariant features", Int. Conf. on Computer Vision, pp. 150-1157, 1999.

[2] Z. Lin and J. Brandt. A local bag-of-features model for large scale object retrieval. In ECCV, 2010. 810.

[3] Zhang L., Chen J., Lu Y., and Wang P., "Face Recogntion using Scale Invariant Feature Transform and Support Vector Machine," in Proceedings of 9th International Conference for Young Computer Scientists, Hunan, pp. 1766-1770, 2008.

[4] von Ahn, L. Human Computation. Doctoral Thesis. UMI Order Number: AAI3205378, CMU, (2005)

[5] Mechanical Turk. http://mturk.com

[6] William Thies, Aishwarya Lakshmi Ratan, and James Davis, Paid Crowdsourcing as a Vehicle for Global Development, ACM CHI 2011 Workshop on Crowdsourcing and Human Computation, May 2011.

[7] Shashank Khanna, Aishwarya Ratan, James Davis, and William Thies, Evaluating and Improving the Usability of Mechanical Turk for Low-Income Workers in India, in ACM DEV 2010.

[8] Aakar Gupta, William Thies, Edward Cutrell and Ravin Balakrishnan. mClerk: Enabling Mobile Crowdsourcing in Developing Regions. ACM CHI 2012 Workshop on Crowdsourcing and Human Computation, May 2012.

[9] Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR, pp. 2161–2168 (2006)

[10] Olivetti Research Labs, Face Dataset, www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[11] Davis, J.; Arderiu, J.; Lin, H.; Nevins, Z.; Schuon, S.; Gallo, O.; and Yang, M.-H. 2010. The hpu. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, 9 –16.

[12] Bernstein, M., Miller, R.C., Little, G., Ackerman, M., Hartmann, B., Karger, D.R., & Panovich, K. Soylent: A Word Processor with a Crowd Inside. Proc. UIST 2010

[13] Little, G., Chilton, L., Goldman, M., and Miller, R.C. TurKit: Human Computation Algorithms on Mechanical Turk. UIST '10, ACM Press (2010).

[14] Kittur, A. and Smus, B. and Khamkar, S. and Kraut, R.E. Crowdforge: Crowdsourcing complex work. In Proceedings of the 24th annual ACM symposium on User interface software and technology, 43—52.

[15] Rodriguez, M. and Davis, J. CrowdSight: Rapidly Prototyping Intelligent Visual Processing Apps. Proc. Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011.

[16] Sorokin, A. and Forsyth, D. Utility data annotation with amazon mechanical turk. Proc. Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on, 1—8.

[17] Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. Proc. CVPR 2009, IEEE (2009), 248-255.

[18] Kittur, A. and Chi, E.H. and Suh, B. Crowdsourcing user studies with Mechanical Turk. In Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems. 453—456.

[19] Bigham, J.P., Jayant, C., Ji, H., Little, G., et al. VizWiz: Nearly Real-time Answers to Visual Questions. UIST '10, ACM Press (2010).

[20] Franklin, M. and Kossmann, D. and Kraska, T. and Ramesh, S. and Xin, R. CrowdDB: answering queries with crowdsourcing. In Proceedings of SIGMOD 2011, 61—72.

[21] Parameswaran, A. and Polyzotis, N. Answering queries using humans, algorithms, and databases. In Proceedings of CIDR 2011.

[22] Turk, M.A. and Pentland, A.P. Face recognition using eigen faces. Proc Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on, 586—591.

[23] Phillips, P.J. and Moon, H. and Rizvi, S.A. and Rauss, P.J. The FERET evaluation methodology for face-recognition algorithms. Proc. Pattern Analysis and Machine Intelligence, IEEE Transactions on, Vol 22, pp 1090—1104.

[24] Zhao, W. and Chellappa, R. and Phillips, P.J. and Rosenfeld, A. Face recognition: A literature survey. Proc Acm Computing Surveys (CSUR), Vol 35, pp 399—458.

[25] Belongie, S. and Malik, J. and Puzicha, J. Shape matching and object recognition using shape contexts. Proc Pattern Analysis and Machine Intelligence, IEEE Transactions on, Vol 24, pp 509—522.

[26] Bay, H. and Ess, A. and Tuytelaars, T. and Van Gool, L. Speeded-up robust features (SURF). Proc. Computer Vision and Image Understanding, Vol 110, pp 346—359.

[27] Narula, P. and Gutheim, P. and Rolnitzky, D. and Kulkarni, A. and Hartmann, B. MobileWorks: A Mobile Crowdsourcing Platform for Workers at the Bottom of the Pyramid. Proc. Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence.

[28] Eagle, N. txteagle: Mobile crowdsourcing. Proc. Internationalization, Design and Global Development, 447—456.

[29] Whitehill, J. and Ruvolo, P. and Wu, T. and Bergsma, J. and Movellan, J. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. Proc. Advances in Neural Information Processing Systems, Vol 22, pp 2035—2043.

[30] Ipeirotis, P.G. and Provost, F. and Wang, J. Quality management on amazon mechanical turk. In Proceedings of the ACM SIGKDD workshop on human computation 2010, 64—67.