To Appear: The 8th IAPR International Conference on Biometrics (ICB), Phuket, Thailand, May 19 - 22, 2015.

Annotating Unconstrained Face Imagery: A Scalable Approach

Emma Taborsky* Kristen Allen* Austin Blanton* Anil K. Jain† Brendan F. Klare*

Abstract

As unconstrained face recognition datasets progress from containing faces that can be automatically detected by commodity face detectors to face imagery with full pose variations that must instead be manually localized, a significant amount of annotation effort is required for developing benchmark datasets. In this work we describe a systematic approach for annotating fully unconstrained face imagery using crowdsourced labor. For such data preparation, a cascade of crowdsourced tasks are performed, which begins with bounding box annotations on all faces contained in images and videos, followed by identification of the labelled person of interest in such imagery, and, finally, landmark annotation of key facial fiducial points. In order to allow such annotations to scale to large volumes of imagery, a software system architecture is provided which achieves a sustained rate of 30,000 annotations per hour (or 500 manual annotations per minute). While previous crowdsourcing guidance described in the literature generally involved multiple choice questions or text input, our tasks required annotators to provide geometric primitives (rectangles and points) in images. As such, algorithms are provided for combining multiple annotations of an image into a single result, and automatically measuring the quality of a given annotation. Finally, other guidance is provided for improving the accuracy and scalability of crowdsourced image annotation for face detection and recognition.

1. Introduction

The past several years have witnessed rapid gains in the development of computer vision algorithms. A large reason for these unprecedented gains is the availability of large scale labelled datasets, which have been critical for the tuning of the millions of parameters needed in deep learning and boosting frameworks. This abundance of available datasets is attributed to two factors. The first factor is the ubiquity of both mobile cameras and social networking sites, which has made the collection and distribution of imagery to populate large scale datasets fairly trivial. The second, and more recent, factor is crowdsourcing services such as Amazon Mechanical Turk (AMT). Together, these two advancements allow for large volumes of imagery to be collected and annotated at orders of magnitude lower cost than was the case a decade ago.

Despite the increased ability to collect and annotate large volumes of data, a well-engineered approach is needed to ensure that the data collected is accurate enough to aid in algorithm development. Without such a measured approach, noisy labels will be prevalent in the data which, in turn, has severe consequences in developing accurate algorithms.

The research discussed in this paper is aimed at improving the accuracy and efficiency of crowdsourced annotations collected during the development of the IARPA Janus Benchmark A (IJB-A) dataset, a large scale, "in the wild", unconstrained face recognition dataset [7]. This dataset contains facial imagery with a lack of constraint in terms of pose, expression, illumination and occlusion. As such, over one million crowdsourced annotations were needed to prepare the dataset for release. A previous paper focused on the dataset details [7], including recognition and detection protocols and baselines, but specific details regarding the annotation process were not covered. As such, this manuscript focuses on providing details on how the data was annotated and key findings from analysis of such annotations to assist others in collecting similar data corpuses.

When collecting such a large number of crowdsourced annotations, two objectives have to be balanced: (i) the accuracy of the crowdsourced annotations are insufficient for use, and (ii) the cost of the crowdsourced annotations becomes as expensive as expert annotation. These two risks are related: redundant crowdsourced annotations are generally needed to improve the accuracy, which in turn increases the cost. The methods described in this work offer an ideal operating point that allows for both accurate and efficient annotation.

The contributions of this work are as follows: (i) a methodology is provided for systematically preparing a fully unconstrained face recognition dataset using crowdsourced labor, (ii) a software system architecture that allows crowdsourced annotations to occur at high throughputs necessary to process large volumes of images and videos (the system processes an average of 500 manual annotations a minute), (iii) algorithms for combining redundant annotations of geometric primitives (rectangular bounding boxes and landmark points), (iv) algorithms for automatically measuring the quality of results provided by annota-

*E. Taborsky, K. Allen, A. Blanton and B. Klare are with Noblis, Falls Church, VA, U.S.A.

†A. Jain is with Michigan State University, East Lansing, MI, U.S.A.

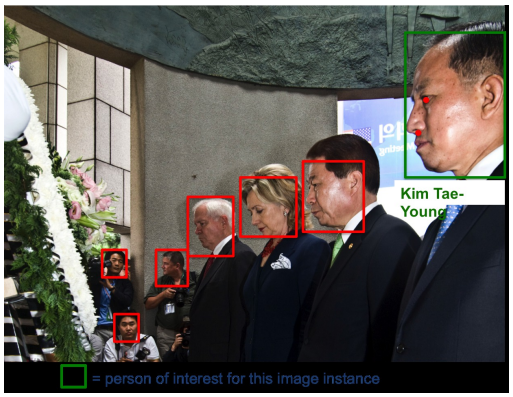


Figure 1. An image annotated from the crowdsourced process detailed in this paper. The final result is the location of all faces, the location of the person of interest (i.e., the labelled subject), and the fiducial landmarks (eyes and nose) for that person. We propose a systematic approach for annotating fully unconstrained imagery at scale.

tors, thus facilitating automated rejection of work and improving overall accuracy of annotation results, and (v) a discussion of other techniques for improving the accuracy and efficiency of crowdsourced dataset preparation.

2. Prior Work

In terms of related works in face recognition, Zhang *et al*. used an automated face detector to collect a database of 2.45 million distinct images, and, leveraging textual information in surrounding web pages, label propagation wass performed to refine label those faces [20]. While similar in motivation, the approach in this paper is noticeably different in that face localization and identity labelling were not performed manually. A related approach was recently used by Yi *et al*. to collect a 500,000 image face dataset [18]. While other face recognition and face detection databases contain manually localized faces [6] or landmarks locations [8], such works seemingly use in house annotators and do not consolidate redundant annotations from weaker crowdsourced workers. One of the most notable uses of crowdsourcing in automated face recognition was Kumar *et al*.'s use of Amazon Mechanical Turk (AMT) to collect attribute labels for face images [9]. Differing from our task where geometric shapes needed to be annotated, the information collected and consolidated by Kumar *et al*. was either multiple choice labels or numeric values, thus simplifying the consolidation process. In the broader computer vision literature, a notable use of crowdsourced annotations include the ImageNet dataset [1], where the binary labelling tasks were performed by AMT workers to verify image labels.

A comprehensive survey of crowdsourcing was conducted by Yuen *et al*. [19]. Similarly, Quinn and Bederson address basic design principles of a crowdsourcing tasking system for quality control with a focus on image-based pattern recognition [11]. Techniques for combining crowd sourced information with low level image features was proposed by Yi *et al*. []. A technique for combining low-level object features with crowd-sourced labels was used to jects to compute the pairwise similarity. Heer and Bostock provide many best practices for crowdsourcing, including the finding that paying workers higher rates results in quicker acceptance of assignments, but not necessarily better quality work [2]. Other related work include a method by Scheirer *et al*. for a framework that helps bridge the gap between human perception and automated detection algorithm [12] through leveraging perceptual information in the form of binary decisions captured from crowdsourced workers. Ipeirotis *et al*. demonstrated that accuracy of redundant crowdsourced tasks begins to saturate at on average, five labels per object, and explored quality measurement on worker output in order to identify error-prone and biased workers [4]. Voyer *et al*. provided a hybrid framework for using both trained analysts and crowdsourced labor by distributing tasks based on skill level required [15]. A previous project created a crowdsourced database of object locations in images by creating a game version of an identification task [14]. Snow *et al*. studied this issue [13] and tested AMT worker responses against an expert's results and determined that, on average, results from four non-experts are comparable to one trained expert. In terms of addressing the efficiency of using crowdsourcing, Mason and Watts [10] demonstrated that utilizing a "quota" scheme, which offers payment for completion of a task, is more cost effective than a "piece rate" approach where workers are paid for the smallest unit of work.

In summary, while crowdsourced data collection is a well studied problem, we believe this to be the first work to provide guidance in crowdsourcing the collection of bounding box and landmark/point locations. Further, while other works performed large scale crowdsourced annotations, guidance for architecting a software system that achieves scale via a high degree of throughput, could not be found. As will be demonstrated, significant challenges exist for each of these approaches.

3. Data development process

3.1. Overview of IJB-A

The IJB-A dataset is a new, publically available dataset [7]. The dataset is a "media in the wild" unconstrained face detection and face recognition corpus, similar in fashion to the LFW dataset [3] and the YTF dataset [17]. However, whereas previous unconstrained face recognition datasets contained face images that were all detected by a commodity face detection algorithm (hence limiting pose variation), a key distinction with the IJB-A dataset is that all faces are manually localized, thus allowing recognition algorithms to be developed on fully unconstrained imagery. While this represents a significant advancement for uncon-

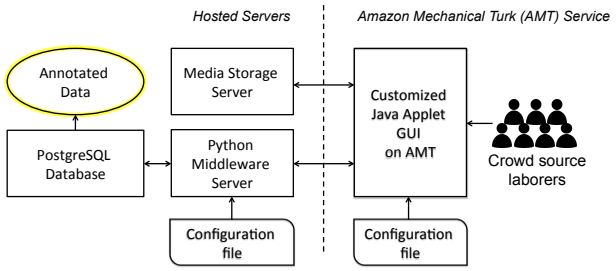


Figure 2. An overview of the software system developed to achieve scalable annotations of facial imagery. Using the Amazon Mechanical Turk (AMT) API, as opposed to the interfaces provided by AMT, allowed for customized GUI for annotations (e.g., bounding boxes and landmark locations), lower costs per HIT, and (most importantly) high annotation throughput (roughly 30,000 annotations per hour can be collected with this system).

strained face recognition algorithm development, it also greatly increases the amount of manual annotation needed to prepare the data for release in the public domain.

The IJB-A database consists of 5,712 images and 2,085 videos of 500 labelled subjects. For each video, the I-frames were extracted, resulting in a total of 20,408 frames. At the time of collection, the image and videos URLs containing a labelled subject were added to a list and then processed by an automated web scraper. For each of the 500 labelled subjects, a "reference image" was also manually collected, consisting of a high-quality frontal face image to help in manual identification of the subject in the collected images and videos. Aside from final inspection, this step was the lone manual task performed in house by "expert" analysts. All the remaining data annotation was performed via crowdsourced annotation tasks.

3.2. Annotation tasks

Given the aforementioned images and videos that were collected, the only information that was known is which labelled subject(s) exist in a given piece of imagery. Because multiple subjects exist in most images and videos, a significant amount of preparation is needed to allow recognition protocols to operate on this data. Specifically, three distinct annotation tasks were performed by crowdsource laborers to facilitate running automated recognition algorithms on this dataset: (i) bounding boxes, (ii) identity verifications, and (iii) landmarks. These tasks occur in order, as each requires information from the previous annotations. Annotations were collected for every image and video frame (I-frames only) in the dataset.

The bounding box task had annotators draw rectangular boxes around all faces in the image (see Figure 3(b)). Specific guidance on marking face boundaries was provided as both written text and images. The result of this task is that the location of all persons with a visible face was known. As such, after this step, the dataset serves as a face detection dataset. While the focus of this paper is on the preparation of a face recognition and face detection dataset, the bounding box task described here also generalizes to most any object recognition task, such as person or vehicle detection.

After the bounding box annotation task, the location of all faces in the image are known; however, it is not known which face corresponds to the labelled subject. As such, the identity (ID) verification task is performed to collect this information. In this task, the workers are shown the reference image for the subject in the given image, and asked to select which face corresponds to that person.

The final task is landmark annotation. In this task key fiducial points on the face of the labelled subject are annotated. For this analysis, data from three landmarks was collected: center of the two eye sockets, and the base of the nose (see Figure 3(d)). This task is important to allow researchers to optionally bypass face alignment steps and focus on face representations and learning algorithms. Similar to the bounding box task, this task could also generalize to other computer vision tasks where specific parts of detected objects could be precisely localized.

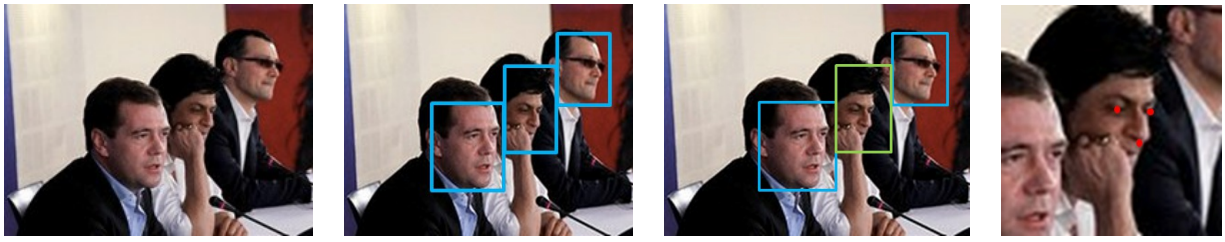
In total, 3,302 annotators worked on this system, with 1,506 annotators working on the bounding box task, 1,010 on ID verification, and 2,100 on landmarks. Workers were paid $\$0.03$ for annotating an image with bounding boxes, $\$0.015$ for ID verification, and $\$0.01$ for landmarks.

3.3. Annotation System Architecture

The annotation framework consists of two separate systems, as illustrated in Figure 2. The first system is the annotation interface which was built on top of the Amazon Mechanical Turk (AMT) service. The second system consists of servers for storing annotations (a relational database), hosting images (a file server), and handling communication between these servers and the AMT service (a middleware server).

For the annotation interface, AMT offers pre-made graphical user interfaces and a "questions" structure for providing Human Intelligence Tasks (HITs) to workers. However, these interfaces could not be used for our tasks because they do not cover required interaction with users (e.g., drawing multiple bounding boxes on an image) and no mechanism exists for automatically passing the results into a relational database. Instead, a Java applet that communicates with the AMT web service API was implemented to allow for such functionality.

The second framework operates in the backend and consists of a system of three servers, which was the key to achieving scale. The first server in this framework is a PostgreSQL relational database. The database contains three primary tables: (i) a table listing all labelled subjects; (ii) a table for all images, which stores all bounding box annotations, and a mapping to the corresponding labelled subject(s) in the image, and (iii) a subject sighting table, which maps specific information in an image to a subject. From these tables, queries are sent to the database to determine



(a) (b) (c) (d)

Figure 3. The three crowdsourced annotation tasks for labelling an unconstrained face detection and recognition dataset. (a) The original image. (b) The "bounding box" annotation task, where workers denote the rectangular regions of the face. (c) The "identity verification" task, where workers are asked to select the face corresponding to the labelled person of interest (show in green). (d) The "landmark" tasks, where specific fiducial landmarks are located (independent of one another, shown in red).

which annotations are missing/needed. In turn, the needed annotations are sent to the annotation system for a human worker to annotate.

As the number of images and annotations grows, the time needed for the database to process a query also increases. At the same time, as the number of workers grows, the number of requests also increases. Combined, these two factors prevent the collection of crowdsourced annotations at large scales as workers must wait for the database system to determine what annotations are needed next. Instead, *a key factor to allow the system to scale is the development of a annotation queue*, where requests for the next annotation task are processed and queued before they actually arrive. Thus, when multiple requests come at the same time, they can be pulled from the queue instead of waiting for database queries.

The second server is the file server which hosts the images and video frames. Technically, this component of the system is straightforward; however, when serving a large number of images to the worker's interface it is imperative to host the file server in an environment that can handle such requests. To this end, our file server is hosted on Amazon's S3 storage platform.

The final server is the middleware server, implemented using Python's Flask framework, which handles all communications between the annotation system and the database and image server using the SQLAlchemy toolkit. The naive approach here is to not use a middleware server, and instead have the Java applet communicate directly to the database system. However, there are multiple benefits of using the proposed middleware server: (i) computationally intensive tasks can be performed on well-specified hardware, as opposed to running client-side on a Turk worker's computer, (ii) updates to the database schema or file server layout do not require any changes to the client front-end, and (iii) database connection information can be hidden from the unknown crowdsourced workers.

To reduce costly communications between the annotation system and the middleware, the requesting of HIT's, in the form of HTTP "GET" requests from the annotation interface to the middleware, and the return of HIT results, in the form of HTTP "PUT" requests, can instead be performed in batch from the client interface. For our system, we found that serving 10 such requests at a time made a significant difference in the throughput of the middleware. Another source of delays in the system can occur when the middleware submits a request to the database; thus, a vital approach is to perform such communications in an asynchronous manner.

Put together, the flow of information through the system is as follows. A crowdsourced worker accepts a HIT through AMT, which causes the annotation applet to be loaded in the worker's browser. The applet sends the middleware server the assignment ID, the middleware sends a request to the database server, the database server pulls the result from the annotation queue, and this is passed back to the middleware. The middleware pulls the corresponding image from the file server, and sends the image and annotation request to the annotation applet. Once the worker completes the annotation, the resultant information is passed back to the middleware, which is then stored in the database. The worker then accepts another HIT and the process repeats.

The result of this system of systems is a presumably unprecedented amount of scale for image-based annotations: *an average rate of 31,500 annotations per hour.* At its peak, the system has processed roughly 46,000 annotations in an hour, which amounts to roughly *10 manual annotations every second.* In total, the system has been used by roughly 3,000 workers, who have completed roughly 1.5 million total annotation tasks. Aside from the overall architectural design, a summary of key approach that allows the proposed system to achieve such scale are: (i) asynchronous database calls, which allows the middleware to serve multiple requests at once, (ii) batch GET's and POST's into groups of n annotations, and (iii) an annotation queue to compute expensive queries in a more paced manner.

4. Consolidating annotations

When utilizing crowdsourcing to complete annotations, three sources of noise exists: (i) systematic biases due to ambiguities in how an annotator interprets a given tasks

(e.g., the exact boundary of a head), (ii) natural variances in the annotators results (e.g., slightly different results if the same annotator annotates the same image twice), and (iii) deliberately malicious annotations (also called spam [5, 16]). Because of these potential sources of error, five to ten users' annotations are collected for each task. In turn, algorithms are needed to consolidate these redundant annotations into a single result. While past work on using crowdsourced annotation has predominantly focused on binary or multiple choice labelling tasks, two of our tasks involve specifying geometric primitives (rectangular regions and points). In this section we describe the methods performed to consolidate each annotation task.

The output of these annotations are considered "inferred truths" (as opposed to ground truths), as they are inferred from redundant, but noisy, annotations. The general approach for conslidation of all three tasks is detecting and excluding outlying annotations (those that vary severely from other users' inputs), followed by an averaging of the remaining annotations.

Bounding box: Bounding box consolidation is the most complex of our annotation tasks, as both the number and location of boxes can vary across annotators. We treat this task as a clustering problem with an unknown number of clusters, as we search for clusters of annotated face regions that infer the location of a face. The clustering process is performed iteratively across each of the annotations for a given image. Let the list L_B consist of entries that are sets of bounding boxes, where each set represents the redundant annotations of a single face in the image. Given n total annotators for an image, the i-th annotation ($1 \geq i \geq n$) will consist of m_i total bounding boxes. Let b_i^j, where $1 \geq j \geq m$, represent the j-th bounding box annotated by person i. An initializer step is performed by populating the list L_B with m_1 sets, each containing a single bounding box b_1^j. Next, for each remaining annotation i', where $2 \geq i' \geq n$, and for each bounding box $b_{i'}^j$, the average overlap is measured between $b_{i'}^j$ and each of the sets in L_B. If the average overlap is greater than τ_O, then bounding box is added to that set. In our system $\tau_O = 0.6$. Otherwise, a new set is created with the single entry $b_{i'}^j$, and added to the list L_B. Finally, after processing all n annotations, the list L_B will consist of sets of overlapping annotations, ideally corresponding to each face in the image. However, due to malicious or erroneous annotations, certain sets in the list may not correspond to a face. As such, a final pruning step is preformed, where sets from L_B are removed if they do not contain at least $\tau_B \cdot n$ boxes, where $\tau_B = 0.6$ in our system. For each remaining set in L_B, the average bounding box is computed and used as the inferred truth for the image.

Identity verification: Identity (ID) verification is a more straightforward annotation task, as it amounts to asking the users to provide an answer to a multiple choice question. Consolidation of ID verification is performed by a simple thresholding: the users' options are "not present in image" or present in one of the image's bounding boxes; if one of the options has a τ_I or more fraction of responses, that answer is accepted as the inferred truth; else an empty inferred truth is stored. In our system $\tau_I = 0.6$. The simplicity of consolidating this annotation task, which is similar to previous crowdsourcing efforts, illustrates the *relative* difficulty of our other two tasks.

Landmarks: Landmarks present a similar challenge as bounding boxes in that location of the annotation points will vary continuously, however only a single annotation is output from an annotator thus removing the notable challenge of a variable number of annotations. Given n annotated points p_i, $1 \leq i \leq n$, the first step is to cluster these points into an unknown number of clusters. The intuition is that after clustering, the true landmark location will correspond to the cluster with a large number of points, and any annotation errors will fall into other clusters. As such, the following iterative procedure is performed. To begin, an empty list L_L is created, whose eventual entries will be sets/clusters of landmarks. This list is initialized by adding the set $S_1 = \{p_1\}$; that is, the first point is added. Next, for each landmark $p_{i'}$, where $2 \leq i' \leq n$, the average normalized distance $d_{i'}^j$ is measured against each of the m sets in L_L as:

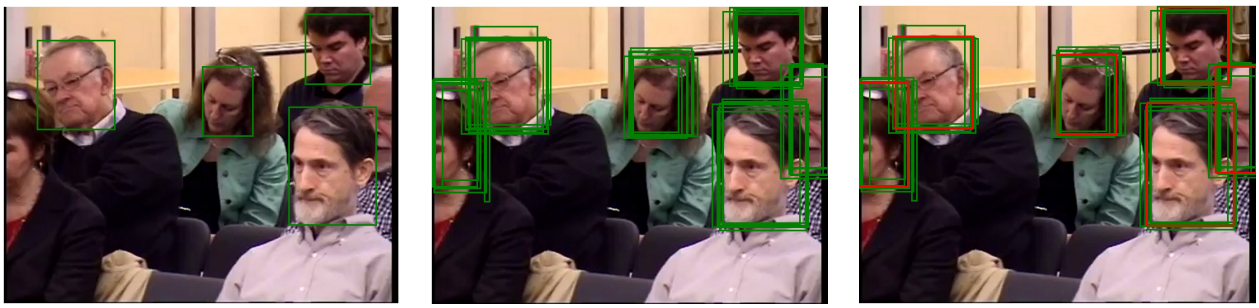
$$d_{i'}^j = \frac{1}{m_j \cdot s} \sum_{k=1}^{m_j} ||S_j(k) - p_{i'}||_2 \quad (1)$$

where m_j is the number of entries in set S_j, $1 \leq j \leq m$, and $s = \max(w, h)$, where w and h are the width and height of the corresponding face bounding box. If $d_{i'}^j < \tau_L$, then $p_{i'}$ is added to set S_j. In the case of multiple $d_{i'}^j$ being less than τ_L, the closest set is selected. Otherwise, a new set $S_{m+1} = \{p_{i'}\}$ is created and added to the list L_L. Finally, points in the set with the most entries, which corresponds to the landmark location, are averaged into the inferred truth p_I. In our system, $\tau_L = 0.05$.

5. Quality metrics

From the inferred truths generated after the consolidation step, the following information can be measured: (i) the quality of a single annotation, (ii) the quality of an annotator, and (iii) the inherent difficulty of an image. In this section we describe how such quality measures are generated. Additionally, we discuss how such measurements can be used to improve the overall accuracy and efficiency of crowdsourced annotations.

Ideally, to determine the quality of incoming annotations, a system would compare and evaluate them against an expertly annotated ground truth. However, this method would obviate the economically efficient collection of unverified annotations through AMT. Instead, we infer an acceptable solution for each annotation from the responses by untrained annotators.



(a) (b) (c)

Figure 4. Our most complex task to crowdsource was bounding box annotation for all faces in the image. (a) An image annotated with bounding boxes by a single worker; some occluded faces are not labelled correctly. (b) Bounding box annotations from all seven workers assigned this image. (c) The results of bounding box consolidation (in red), where all the faces have been properly annotated.

5.1. Annotation quality

The annotation quality metric evaluates the accuracy of a single annotation. The quality score for each of the three annotation tasks is measured in a similar manner: measuring difference between a given annotation and the inferred truth computed from the consolidation step.

Bounding Box: The bounding box quality score for an image is computed by first measuring the overlap between each inferred truth bounding box and the list of bounding boxes from an annotator, where the maximum overlap is computed for each inferred truth box. Thus, with m inferred truth bounding boxes, the result for the i-th annotations is a vector of overlaps $v_i \in \mathbb{R}^m$, where $0 \leq v_i(j) \leq 1$, and $1 \leq j \leq m$. Because slightly overlapping bounding boxes can be more problematic than missing bounding boxes, they are treated with the same penalty score: a score of 0. Thus, the final bounding box quality score q_B is computed as:

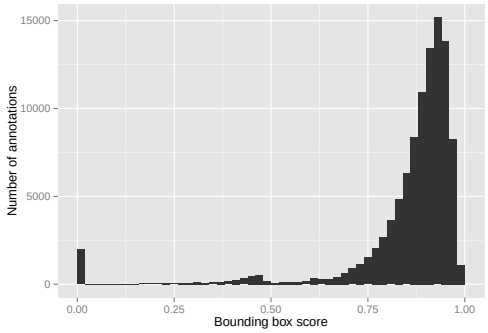
$$q_B = \frac{1}{m} \cdot \sum_{j=1}^{m} f_b(v_i(j)) \tag{2}$$

where $f_B(\cdot)$ is computed as:

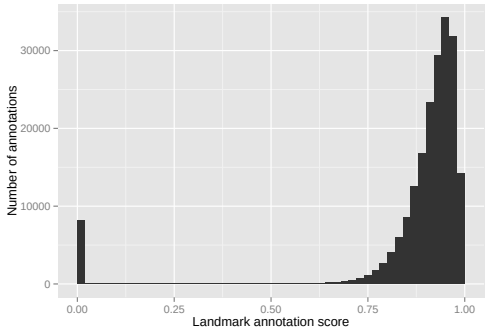
$$f_B(v_i(j)) = \begin{cases} v_i(j) & : v_i(j) \geq \tau_O \\ 0 & : v_i(j) < \tau_O \end{cases} \tag{3}$$

See Section 4 for discussion on τ_O. Figure 5 shows the distribution of a sample of 104,093 quality scores from the bounding box task. The shape of the distribution supports the existince of systematic biases in labelling (peak is not at 1.0), variance in annotators ability (spread of the distribution), and malicious annotations (a second peak at 0.0).

Identity Verification: Quality score for the identity (id) verification task is straightforward: either the annotation matches inferred truth, in which case a score of 1.0 is assigned, otherwise it is incorrect, and a score of 0 is assigned. From a sample of the 187,530 ID annotations, 93.6% were correct (i.e., received a quality score of 1.0). The simplicity of this quality metric again illustrates the difference in difficulty when using crowdsourcing to collect multiple choice answers versus geometric primitives (in the case of the bounding box and landmark tasks).



(a)



(b)

Figure 5. Distribution of quality scores for (a) the bounding box task, and (b) the landmark annotation task. Each of these distribution is bi-modal: the peaks at 0.0 are indicative of malicious annotators. The main distribution peaks at a score slightly lower than 1.0, which indicates biases from the annotators, and has a natural variance which indicates expected noise in the annotation process.

Landmarks: When computing landmark annotation quality score, we first measure the distance d_L, in pixels, as $d_L = ||p_A - p_I||_2$, where p_A is the annotated point, and p_I is the inferred truth point. Let $s = \max(h, w)$, where h

is the height, in pixels, of the face bounding box, and w is the width. The landmark quality score q_L is defined as:

$$q_L = \begin{cases} 1 - \frac{d_L}{2 \cdot s \cdot \tau_L} & : \frac{d_L}{s} \leq \tau_L \\ 0 & : d_L < \tau_L \end{cases} \quad (4)$$

The quality metric first thresholds scores that are beyond τ_L distance from the inferred truth to be 0. Next, the distance is linearly scaled such that a max distance of τ_L results in a quality score of 0.5 and a distance of 0 results in a score of 1.0. As seen in Figure 5, the landmark quality metric offers a similar distribution in terms of a biases peak, a natural variance, and a second peak at 0 indicating potentially malicious annotations.

Together with the consolidation algorithms provided in Section 4, the quality metrics can be used to iteratively improve the accuracy of the inferred truths. That is, after first consolidation, and then measuring annotation quality, annotations that score below a given threshold can be removed from the system. In turn the consolidation process can be run again, this time containing fewer noisy annotations. In practice, generally only two iterations are needed for this approach to converge.

5.2. Annotator quality

Annotator quality score is defined as a per-task average of that user's annotation scores: thus, a user who participates in each of our tasks will have a score for bounding boxes, a combined score for landmarks and a score for id verifications. The annotator score for a given task is simply the average of quality scores of his annotations. Computing annotator average scores allows for the identification and purging of malicious annotators.

5.3. Image difficulty

Certain images are inherently more difficult to annotate than others. Being able to identity such images allows resources to be dynamically allocated in a manner that assigns fewer workers to easy images and additional annotators to more difficult images. In this section we provide an algorithm for measuring the inherent difficulty of an image which uses the information from the consolidation algorithms provided in Section 4.

The proposed approach to measuring image difficulty is to measure the level of disagreement amongst annotators. Notably, our consolidation algorithms discussed in Section 4 implement different clustering approaches to combine correct annotations together and exclude bad annotations. The results of these clustering approaches are stored in this L_B for bounding box, and L_L for landmarks. Let n refer to the number of annotations performed on a given image.

For the bounding box task, sets of bounding boxes in the list L_B were accepted as face locations if the set contained $\tau_B \cdot n$ entries. Thus, given c sets that exceed this threshold, the bounding box image difficulty D_B is defined as:

$$D_B = 1 - \frac{c}{|L_B|} \quad (5)$$

where $|L_B|$ is the number of entries in list L_B.

For id verification, the image difficulty is simply the percentage annotators who selected the correct answer. Thus, when unanimous decisions do not occur, the image will be measured as being more difficult.

For the landmark annotation task, sets of landmarks are stored in L_L. Here, only one of the sets is accepted as inferred truth result. Thus, the landmark image difficulty D_L is defined as:

$$D_L = 1 - \frac{1}{|L_L|} \quad (6)$$

An example of images or video frames that were measured as having low difficulty ($D_B < 0.05$) can be found in Figure 6(a), and imagery measure as having high difficulty ($D_B > 0.8$) can be found in Figure 6(b). Such a qualitative analysis of the remaining imagery with respect to difficulty score demonstrated the ability of this method for detecting challenging imagery. The implication is that the most difficult images can be flagged for annotation by an expert annotators. Other images can receive a dynamic number of annotators. For example, after collecting roughly four annotations, we can automatically determine whether or not to send an image out for additional crowdsourced annotations, or to send to an expert.

6. Conclusions

A systematic approach was provided for annotating a fully unconstrained face dataset using crowdsourced workers. As face recognition continues to saturate on seminal datasets such as LFW and YTF, a higher degree of manual labelling (bounding box for faces and facial landmarks) of imagery will be required to push the state of the art. As such, details were provided on how to systematically decompose and delegate such tasks and how to implement annotation systems that can scale to allow hundreds of crowdsourced workers to work simultaneously. At the same time, as more complex annotation tasks are required, such as annotation of fiducial landmarks or placement of bounding boxes around objects such as faces, the consolidation of the redundant annotations required when using crowdsourced labor becomes more challenging. Thus, algorithms were provided for both consolidating such annotations, and automatically measuring worker quality. In our annotation task for the IJB-A face dataset, a total of 3,302 crowdsourced workers were employed to annotate 26,120 images and video frames. The total number of annotations were 1,600,443 at a total cost of $23,120.03. With the described system in place, data annotation can be continually repeated at high throughputs and low cost. Future work will focus on further automating the system and performing additional analysis of annotation results.

(a) (b)

Figure 6. Examples of images that were measured as being inherently (a) easy and (b) difficult to annotate for the bounding box task using the algorithms described in Section 5.3. The easiest images typically involve clearly visible faces without any occlusions. Difficult images generally involved large crowds, lower resolutions persons in the background, or ambiguous cases such as pictures on the wall or persons on television screens.

Acknowledgement

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA) under contract number 2014-14071600010. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright annotation thereon.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. of CVPR*, pages 248–255, 2009.

[2] J. Heer and M. Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *SIGCHI Human Factors in Computing Systems*, pages 203–212, 2010.

[3] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[4] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on Amazon Mechanical Turk. In *SIGKDD Workshop on Human Computation*, pages 64–67, 2010.

[5] S. Jagabathula, L. Subramanian, and A. Venkataraman. Reputation-based worker filtering in crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 2492–2500, 2014.

[6] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

[7] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, , and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In *Proc. of CVPR (to appear)*, 2015.

[8] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *IEEE Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.

[9] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Describable visual attributes for face verification and image search. In *IEEE PAMI*, volume 33, pages 1962–1977, October 2011.

[10] W. Mason and D. J. Watts. Financial incentives and the performance of crowds. *ACM SigKDD Explorations Newsletter*, 11(2):100–108, 2010.

[11] A. J. Quinn and B. B. Bederson. Human computation: a survey and taxonomy of a growing field. In *SIGCHI Human Factors in Computing Systems*, pages 1403–1412, 2011.

[12] W. Scheirer, S. Anthony, K. Nakayama, and D. Cox. Perceptual annotation: Measuring human vision to improve computer vision. *IEEE PAMI*, 36(8), 2014.

[13] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Empirical Methods in Natural Language Processing*, pages 254–263, 2008.

[14] L. Von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. In *SIGCHI Human Factors in Computing Systems*, pages 55–64, 2006.

[15] R. Voyer, V. Nygaard, W. Fitzgerald, and H. Copperman. A hybrid model for annotating named entity training corpora. In *Linguistic Annotation Workshop*, pages 243–246, 2010.

[16] J. B. Vuurens and A. P. de Vries. Obtaining high-quality relevance judgments using crowdsourcing. *IEEE Internet Computing*, 16(5):20–27, 2012.

[17] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *Proc. of CVPR*, pages 529–534, 2011.

[18] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. In *arXiv:1411.7923*, 2014.

[19] M.-C. Yuen, I. King, and K.-S. Leung. A survey of crowdsourcing systems. In *IEEE Privacy, Security, Risk and Trust*, pages 766–773, 2011.

[20] X. Zhang, L. Zhang, X.-J. Wang, and H.-Y. Shum. Finding celebrities in billions of web images. *IEEE Multimedia*, 14(4):995–1007, 2012.