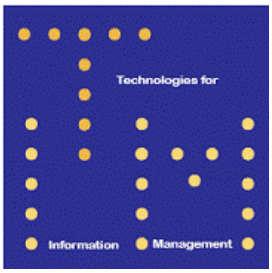
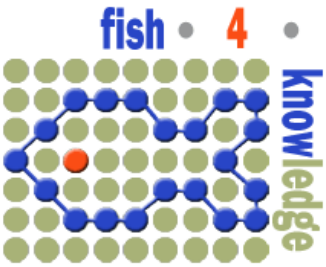




Fish4Knowledge Deliverable 5.6

Video Ground Truth Generation

Principal Author: Bastiaan J. Boom, Concetto Spampinato, Jiyin He, Emmanuelle Beauxis-Aussalet, Isaak Kavasidis
Contributors: UEDIN
Dissemination: PU

Abstract:

Deliverable due: 3 Month

1 Introduction

For the evaluation, multiple groundtruth annotation interfaces have been developed in order to obtain data that allows us to evaluate the image processing software. Without this data, the evaluation of the components is impossible, but in most cases obtaining good quality annotations is difficult. In the Fish4Knowledge project, multiple tasks in video/image processing like fish detection, fish recognition and behaviour classification are necessary to analyse the data. This however also requires different kind of interfaces for annotating the requered data for each task. In this section, a summary of the different interfaces for annotating the data is given, a more detailed description for each interfaces is provided in the next sections.

1. Perla (fish detection): This is a web interface for labeling the contour and trajectory of the fish in the video. An example of this web interface is shown in the top of Figure 1. It allows multiple people to annotate the trajectory and the contour of the fish and later combine those annotations. (Section 2)

2. Flash the Fish Game (fish detection): The fish game (middle-left of Figure 1) is a fun way to perform the annotation of fish, where the annotator plays a diver in the game with a camera that has to take pictures of the fish. These picture allow us to define the location of the fish in the video. Notice however that these annotations do not give a contour. (Section 3)

3. Fish behaviour (fish behaviour): For the fish behaviour, an annotation website (middle-right of Figure 1) is created which allows users to search for combinations of species in the videos, for instance if two clown fish appear in the video around the same time. Afterward, we can annotate if these fish are interacting with each other in certain way, for instance pairing. (Section 4).

4. Clustering interface (fish recognition): A website (bottom-left of Figure 1) is created to annotate the fish species, where we first remove the species that are incorrectly classified to that cluster and afterwards link this cluster to a certain species. This allows users to annotate fish images $3\times$ faster than annotating each image separately. It even makes the annotation task simpler so no domain knowledge is required. (Section 5)

5. Fish labeling game (fish recognition): This interface (bottom-right of Figure 1) transforms the difficult task of recognising fish species into an easier game task that only requires visual similarirty judgements. (Section 6)

2 Perla

2.1 Collaborative Environments and Crowdsourcing for Ground Truth Generation

Because groundtruth generation is a fundamental task in the design and testing of computer vision algorithms, in the last decade the multimedia and, more in general, the computer vision community has developed a disparate number of annotation frameworks and tools to help

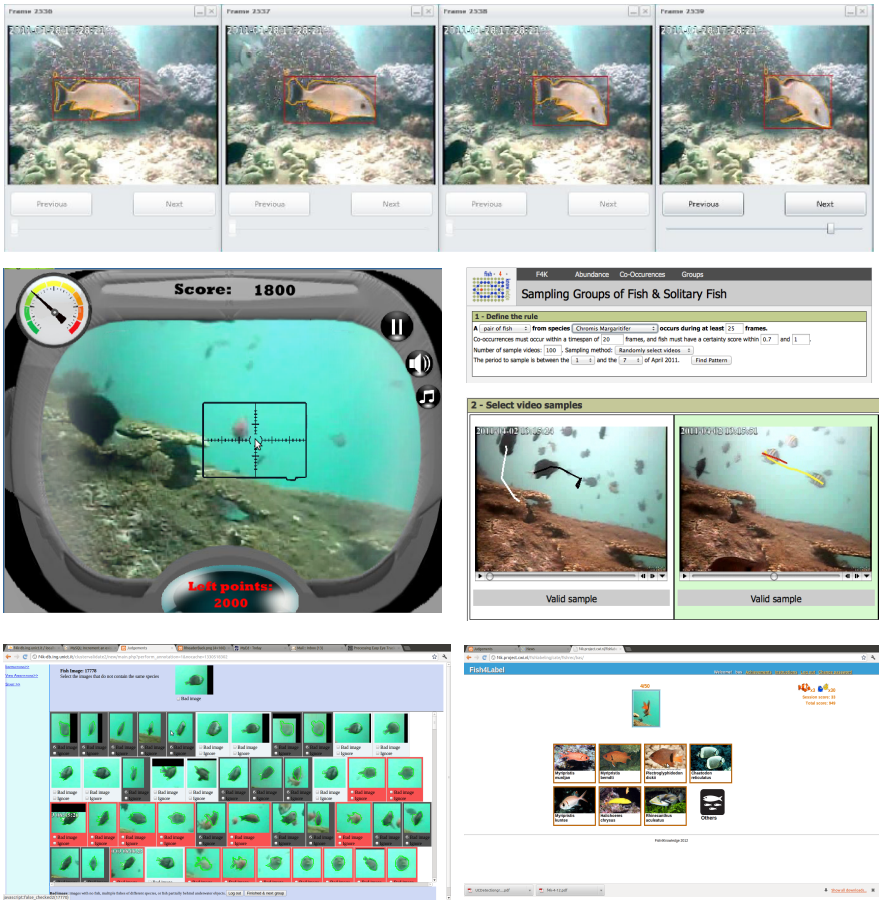


Figure 1: Examples of interfaces that have been developed for annotation of image processing groundtruth data

researchers in collecting datasets, which are then used in the tasks of image segmentation, object detection and tracking, object recognition, etc.

The most common approaches are devised as stand-alone tools created by isolated research groups, and as such, tailored to specific needs. These include, for instance, ViPER-GT [7], GTVT [2], GTTool [13], ODViS [12], which, however, show their limitations when it comes to generate large scale groundtruth datasets. In fact, they exploit the efforts of a limited number of people and do not support sharing of labeled data. All these needs combined with the rapid growth of the Internet have favored in the last years the expansion of web-based collaborative tools, which take advantage of the efforts of large groups of people to collect reliable groundtruths. LabelMe [16], a web-based platform to collect user annotations in still images, is a significant example. However, LabelMe lacks intelligent mechanisms for quality control and integration of user annotations. In fact, quality control is achieved by a simple approach that counts the number of annotation landmarks, and it does not exploit the full potential of its collaborative nature (being a web-based platform) since annotations of multiple users of the same object instance are not combined. In fact, the LabelME dataset, though being one of the largest datasets available, it is particularly inaccurate. Moreover, LabelMe is designed specifically for still images, although a video based version has been proposed [21] that, however, is not as successful and flexible as the image based version.

Sorokin and Forsyth [17] have, recently, demonstrated the utility of "crowdsourcing" to human resources (non-experts) the task of collecting large annotated datasets. Nevertheless, two main aspects have to be taken into account when crowdsourcing: workers' motivation and control. The easiest and most natural way to motivate people is paying them for their work. This strategy is applied by Amazon's Mechanical Turk service [15] and CrowdFlower [4]. A valid alternative for workers motivation is personal amusement [18]: this is the case of the ESP and Peekaboom games [1] which exploit players' agreement (randomly pairing two players and let them guess each other's labels) to collect groundtruth data.
Beside workers motivation, another concern of crowdsourcing solutions is the quality control over annotators, which has been tackled with different strategies that can be summarized [**?**] as: Task Redundancy (ask multiple users to annotate the same data), User Reputation and Groundtruth seeding (i.e. coupling groundtruth with test data). Although these solutions are able to build large scale datasets, they might be very expensive and contain low quality annotation since workers (even if payed) are not as motivated as researchers.
There exist also approaches, which try to generate groundtruth data automatically (without human intervention) video and image data [10, 3] but they rely on approaches that cannot be fully trusted.
For all the above reasons, we have developed two approaches that involve humans and algorithms in the labeling process: the first one, *PerLa* that is a web-based collaborative tool for generating hand-labeled object detection, tracking and recognition groundtruth supported by image processing algorithms and, the last one, *Flash the Fish*, which is an online game allowing us to collect large scale groundtruth while people playing it. Both approaches share the same database schema, shown in Fig. 2 to store the annotated data. The philosophy is that for each video we have many *ground_truth*s generated by multiple users and each *ground_truth* contains many *object*s. The *ground_truth*s of each video are then combined in one *best_ground_truth* which contains a set of *best_object*s resulting from the integration of the *object*s generated by all the users who have annotated the video under consideration.

2.2 PERformance evaluation, Labeling and Annotation tool

PERLA (PERformance evaluation, Labeling and Annotation) tool is a collaborative environment that allows users to create and share their video annotations, thus accelerating the generation of high quality video groundtruth by increasing/integrating the number of annotations in a sort of inherent user supervision. It is described in Deliverable 2.3. The proposed tool has been adopted for groundtruth data collection within the Fish4Knowledge project, whose video repository holds more than half a million videos at different resolutions and frame rates.
At the date of January 31, 2013, the PERLA database contains 55 annotated videos with 55332 annotations (about 2900 different objects) in 24136 video frames, collected by several users, which is online since July 01, 2012, though it has not been fully advertised.
Fig. 3 shows the histogram of the total number of annotated images with respect to the percentage of labeled pixels. In particular, 10034 frames have less than 10% of pixels labeled and no image has more than 60% of pixels labeled. The histogram of the number of images per the number of objects in these images (see Fig. 4), instead, shows that there exists a high number of images with only one annotation (a little more than 11000).

Currently, the tool's database is constantly growing, since more and more new users are working on the annotation of new image sequences. At the current rate, we estimate about 150

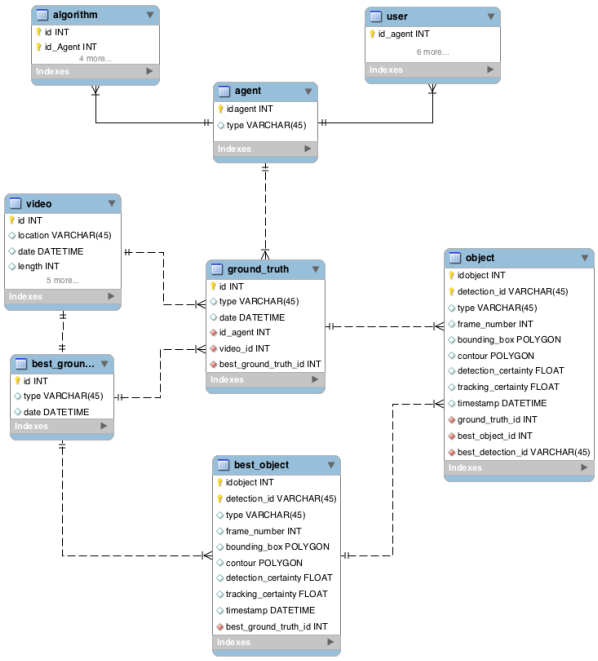


Figure 2: Ground Truth DB Schema.

10-minute videos annotated by the end of 2013, resulting in about 150.000 annotations of about 10.000 different objects.

3 Flash the Fish

Ground truth generation is a tedious task, and annotators must be highly motivated to finish their work in a reasonable amount of time and with good quality. Besides money, an effective strategy to motivate people is amusement and *Flash the Fish* adopts this strategy to generate large scale object detection groundtruth.

The game is quite simple: the user is presented a segment of an underwater video and the only thing that he/she has to do is taking photos of the fish, by clicking on them (Fig. 5) gaining as many points as possible. The user needs to gather a certain score to get next game levels. Each shot "photo" contributes in estimating the presence or absence of fish at the corresponding point in the video.

Currently, the game consists of 8 different levels of progressively increasing difficulty. The first level serves the role of assessing the skills of the player (see next section) and has an initial frame rate of 5 FPS and the time available is 35 seconds. At each successive level the frame rate of the video segment is increased by one, while the time available is reduced by 2 seconds, to a maximum of 12 FPS and a minimum of 21 seconds at the 8th and last level. The game is available at *http://f4k-db.ing.unict.it/*.

In order to make the game more appealing, we adopted a scoring system that rewards users according to the quality of their annotations. In other words, the more precise the user is, the more points he/she earns and climbs up the final classification. Of course, in order to be able to assign scores, it is necessary that each video segment comes with a reference groundtruth. If, for the specific video, there exists a hand-made groundtruth, it will be used. Otherwise, if the video

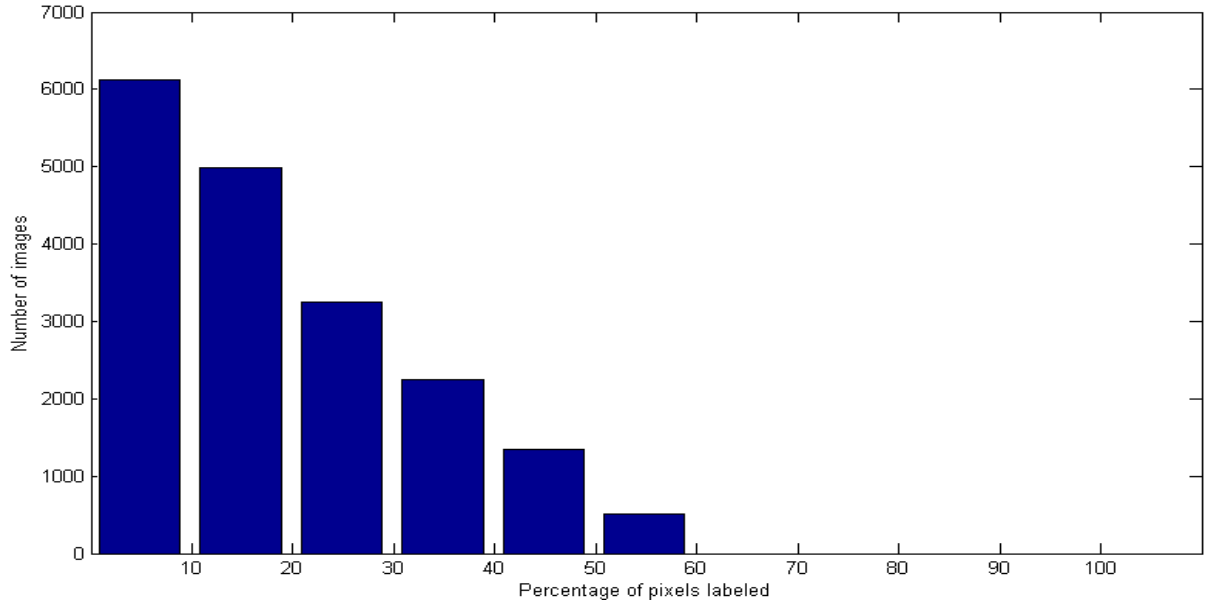


Figure 3: Histogram of the number of images with respect to the pixel coverage.

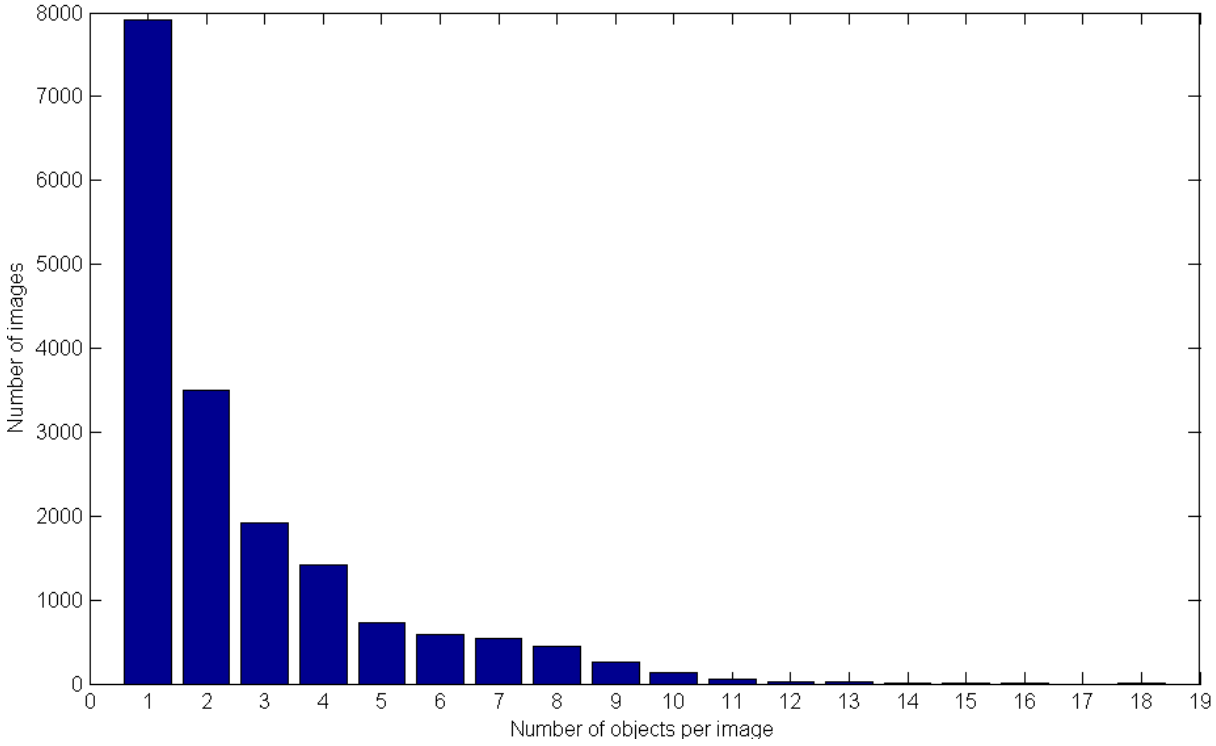


Figure 4: Histogram of the number of images with respect to the number of objects present.

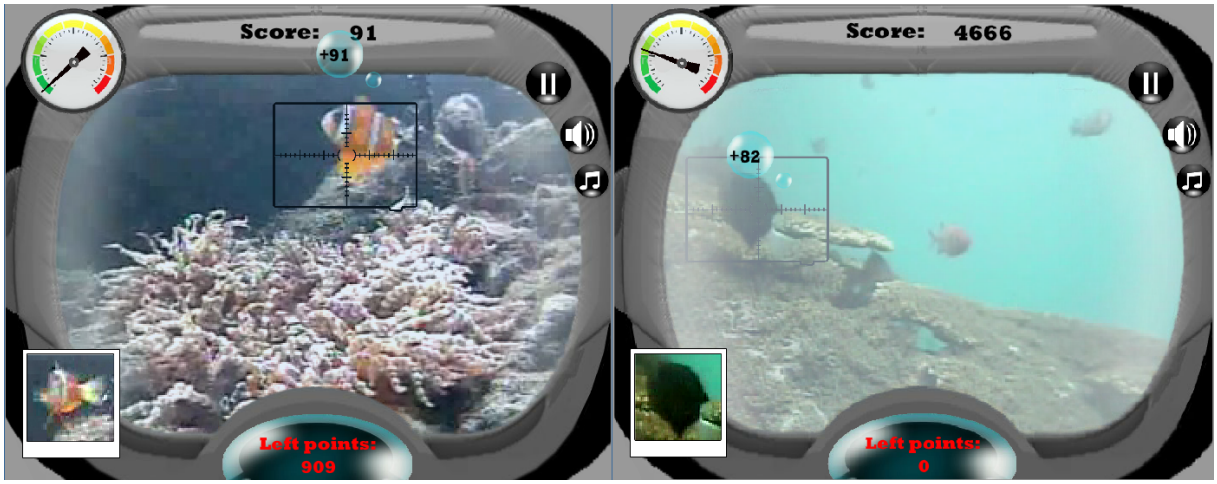


Figure 5: The game's interface.

is not a new one (i.e. several players have already played it, meaning that several annotations exist), the reference groundtruth is given by the combination of all the existing annotations (see next section). If, instead, the video is a new one (i.e. no one has played it yet) then the detection algorithms' output is used as reference groundtruth.

Having a reference groundtruth, it is possible to compare the annotations provided by the users against it. For each object in the reference groundtruth a 2D Gaussian Distribution is placed, centered on the object's bounding box center. If a player clicks on that point, he/she gains the maximum points he/she can get, while the points awarded are reduced as the clicked point gets more distant from the center.

3.1 Data Analysis and Integration

In order to make sense of the data produced by this game, we had to deal with the following issues:

1. **Assess the quality of the users:** The contribution of each user playing the game cannot be equal. In fact, there exist casual players that dedicate a little time playing, achieving, usually, low scores and on the other extreme, hardcore players that are able to memorize every single detail of a game, can be found. Assessing user quality is of key important for generating a groundtruth based on the weighted contribution of users' contributions. The weight is the quality score itself, meaning that the higher a player's score is, the more influential his/her annotations will be in determining the final groundtruth.
   To estimate user quality we resort to groundtruth seeding, i.e. the first level of the game contains a video for which a reference groundtruth (G_{GT}) already exists. When the first level of the game ends, the acquired data (GT_u) of the user u is compared to the G_{GT}. Each submitted groundtruth starts with a quality score (S_{GT}) of 1 and the number of False Positives (FP_u, a location where the user clicked but fish does not exist), False Negatives (FN_u, a location where the user did not click but fish does exist) and True Positives (TP_u, a location where the user clicked and fish does exist) are determined. While a TP_u does not decrease the quality of the groundtruth and a FP_u decreases it always, a FN_u is more complicated because it can occur for two reasons: 1) the user did not click on it

at all, because he/she was not fast enough, or 2) because, at the same time, he/she was clicking on another fish. In the former case, if the user was not fast enough to click, S_{GT} is decremented by N_{ft}/N_d, where N_d and N_{f_t} are the objects contained, respectively, in the G_{GT} and in the frame f_t. If the user was clicking other objects at the time that FN_u occurred, is determined by seeking for objects in frame f_t. If such objects exist, and they were shot by the user, no action is taken. Conversely, the user's quality is decremented as before.

Summarizing the score of each submitted groundtruth is given by:

$$S_{GT} = 1 - \frac{1}{N_d} \sum_{N_d} N_{false} \tag{1}$$

where

$$N_{false} = \begin{cases} 0, \text{if } ClickedLocation \text{ is a } TP_u \text{or is a } FN_u \text{ and } \exists\, TP_u \text{ in the same frame} \\ 1, \text{is a } FN_u \text{ and } \nexists\, TP_u \text{ in the same frame} \end{cases}$$

If this is the first groundtruth created by the user, his/her quality score is equal to S_{GT}. If, instead, previous assessments exist, the quality score of the user is determined by:

$$S_u = \frac{1}{N_{Tot}} \sum_{i=1}^{U_{GT}} S_{GT_i} \times N_{GO_i} \tag{2}$$

where N_{Tot} is the number of objects in all the groundtruths of the user, U_{GT} is the set of his/her groundtruths, S_{GT_i} is the quality of i^{th} groundtruth, given by (1), and N_{GO_i} is the number of objects in it.

2. **Build the groundtruth objects:** Once the users obtain a quality score, their annotations can be integrated in order to build the best groundtruths. In order to identify the locations that users clicked the most, we apply iteratively an unsupervised K-Means algorithm. In particular, this algorithm starts with a predefined number of clusters (set to 10 or to the number of fish in the existing groundtruth, if it contains more) and then iterates through each point (clicked by the user) and determines whether it fits well in the assigned cluster or not. If it is positive (correct assignment) or negative (wrong assignment, but fits neighboring cluster), the point is marked as confirmed and it will be included in the next iteration. On the contrary, if the point's silhouette value lies near zero, it is removed (unassigned) from the cluster and is excluded from successive iterations.
   At each iteration, every cluster c is assigned a value that represents their significance, or their radius, and is given by:

$$r_c = \frac{1}{N} \sum_{p}^{P_c} Q_{u,p} \tag{3}$$

where N is the total number of points in the current frame, p represents the points in that cluster and $Q_{u,p}$ is the quality of the user that created that point.

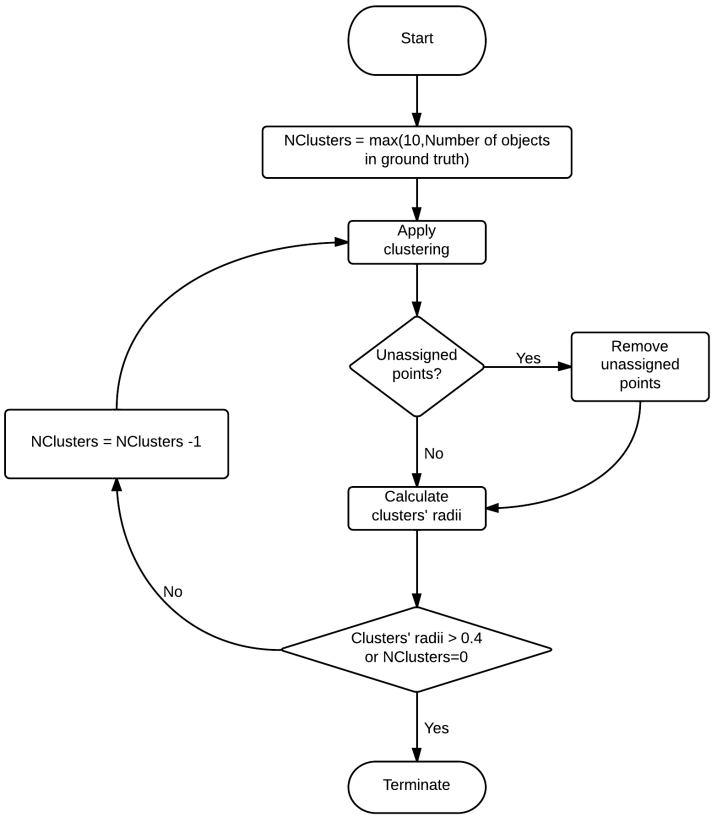


Figure 6: The clustering algorithm.

The algorithm stops when all the remaining clusters have a value of $r_c > Th$ (Th empirically set to 0.4) or there are no starting clusters available. In case these conditions are not satisfied, the starting cluster number is decreased by one and the algorithm proceeds with the next iteration. The resulting clusters can be represented as heatmaps, enabling us to identify how the users' clicks are distributed. Fig. 6 shows the flowchart of the clustering algorithm, Fig. 8 shows an example output of the method described and Fig. 7 shows the heatmaps produced in a frame sequence.

While it is possible that many TP_us are discarded during this step, because valid positions are excluded or the clusters are too small ($r_c < Th$), their influence will progressively increase as more annotations stack up.

Flash the Fish is publicly available since November 1st 2012. At the date of January, 31st 2013, 65 different users played the game creating more than 1200 groundtruths that contain about 210000 fish.

3.2 Work in Progress: Bonus Levels

We are currently working on two bonus levels that the user can play in order to create groundtruth, respectively, for tracking and recognition algorithms. The tracking groundtruth generation level (Fig. 9, left) shows a video segment with a fish moving and asks the user to follow it with his/her mouse. By following this approach we are able to map the user's movements to the trajectory of

Figure 7: Heatmaps of two fish detected in an 8-frame sequence.

Figure 8: Clustering applied on the acquired data: Red dots are the locations clicked by the users. Yellow circles represent the result of the first clustering iteration, while the blue circles are the final result of the clustering method. The radius of each circle is equal to the sum of the quality scores of the users that made an annotation that belongs to that cluster, given by (3).

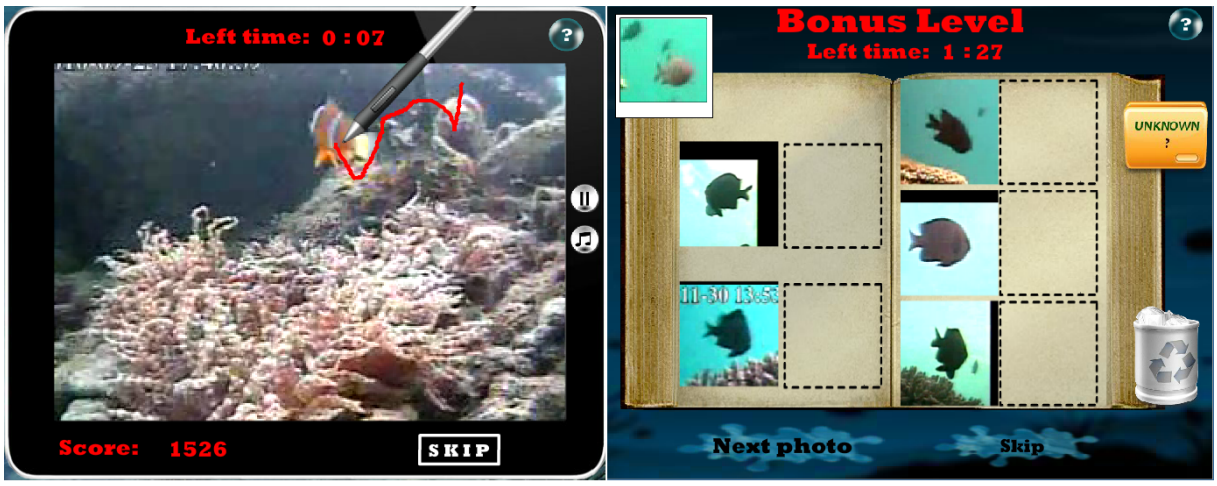


Figure 9: The bonus levels. *Left:* In the tracking bonus level the user has to follow a fish by drawing its trajectory. *Right:* In the recognition level the user has to drag and drop the upper left corner image to the one, from the images below, that it matches the best.

the fish. The score is given by matching the trajectory of the fish and the line drawn by the user. In addition, we are currently working on deriving tracking information directly by processing the sequence of the taken photos.
The recognition groundtruth generation level (Fig. 9, right), instead, shows a photo of a fish (the reference image), for which its features are already known by the system, on the upper left corner, and 5 photos taken in the previous level. The user is asked to indicate which one of the unidentified fish is the same as the reference one, by selecting one of them. The score is computed by matching the SIFT keypoints of the selected photo with the ones of the reference image.

4 Annotating fish behaviour

4.1 Overview of the interface

The user interface developed for behaviour annotations addresses two main concerns:

- **Handling the specification of meaningful events**: We support the user-defined interpretation of fish interactions. For instance, groups of fish can gather for reproduction activities or for feeding activities, depending on the species.
- **Reducing the effort needed to collect training datasets** - The collection of training datasets is a tedious and time-consuming task. It involves filtering, browsing and watching numerous videos. For instance most of the videos may not contain any occurrence of the event of interest, thus being irrelevant for users.

We based the specification of meaningful events on the user study conducted for the Fish4Knowledge project[1]. End-users expressed interest in fish interactions related to demographics, reproduction,

[1] http://homepages.inf.ed.ac.uk/rbf/Fish4Knowledge/DELIVERABLES/Del21.pdf

Fish Species	Solitary	Pairing
Dascyllus Reticulatus	Abnormal	Breeding
Chromis Margaritifer	Normal	Breeding
Plectrogly-Phidodon dickii	*unknown*	Breeding
Acanthurus Nigrofuscus	Abnormal	*unknown*
Pomacentrus Moluccenis	Abnormal	Breeding
Chaetodon Trifascialis	Normal	Normal Breeding
Zebrasoma Scopas	Juvenile	Rare
8 Scolopsis Bilineate	Juvenile	Adult
Amphiprion Clarkii	*unknown*	Breeding
Siganus Fuscescens	Abnormal	*unknown*

Table 1: Interpretation of Solitary and Pairing Events depending on Fish Species

feeding, and environmental conditions. They elicited 10 species that are the most interesting to study because their behaviours are representative of the ecosystem conditions. We derived the specific fish behaviours of interests on the basis of descriptions of the 10 species provided by end-users and by the FishBase project[2].

We specifically focused on pairing and solitary behaviours, as they address biologists' interests in demographics, reproduction, feeding, and environmental conditions. The meaning of pairing and solitary events depend on the species involved, and the Table 1 summarizes their interpretation.

To reduce the effort needed for collecting training datasets, we designed a rule-based interface. The rules support users in retrieving the video that have a high chance of containing an example of the event of interest. The rules basically define the fish co-occurrences to retrieve in the videos.

As shown in Fig.10, the rule parameters that define the events of interest are: species of interest, number of co-occurring fish, maximum delay between each fish occurrences, and minimum duration of co-occurrences. These parameters refer to the video data produced by the Fish Detection, Fish Tracking and Fish Recognition components. These components supply the data needed for retrieving the videos that satisfy the user-defined rules.

The Fig.10 also shows that users can apply specific sampling methods: they can select the time periods to sample, randomize the ordering of the retrieved samples, and specify the number of samples needed.

In this way, the UI handles the specification of meaningful events and reduces the effort needed to collect ground-truth datasets. In particular, it achieves the following points:

- The effort needed to define the rule parameters is reduced to a limited number of form inputs to fill in, and user inputs are integrated in human-understandable sentences.
- The rules are sufficiently flexible to address the set of events of interests from Table 1.

[2]http://fishbase.org

Figure 10: Screenshots of user-defined rules for retrieving solitary and paring fish (first two images), and for retrieving co-occurrences of 2 species (last image).

Fish Species	Behaviour	Rule
Chromis Margaritifer	Solitary	N=1, S=2, F=30, T=35
Chaetodon Trifascialis	Solitary	N=1, S=6, F=10, T=10
Scolopsis Bilineate	Solitary	N=1, S=8, F=25, T=25
Dascyllus Reticulatus	Pairing	N=2, S=1, F=10, T=25
Plectrogly-Phidodon dickii	Pairing	N=2, S=3, F=5, T=20
Pomacentrus Moluccensis	Pairing	N=2, S=5, F=10, T=5

Table 2: Example of rules used for event detection. N stands for number of co-occurring fish (N=1 meaning one fish co-occurs with no other fish), S for species of interest, F for the number of frames in which fish co-occur, and T for the timespan between each fish occurrence.

The behaviour annotation system was used for collecting ground-truth datasets containing the behaviours described in Table 1. But such a tool can be used to target a wider range of fish behaviors. For instance, this can be done by using the rules developed to target co-occurrences of fish from different species, and occurrences of groups of fish from the same species.

4.2 User interactions

The user interface functionalities support i) the retrieval of video excerpts that display the co-occurrences of interest, and ii) the manual selection of video excerpts that are suitable for the training dataset. It organizes the dataset collection task in 3 steps:

1. **Define the rule, and the sampling method**
   Users are supported with 2 simple rules, and a set of parameters they can modify. The most important rule supports the retrieval of solitary fish and pairing fish. It covers most of the events of interest from Table 1. An additional rule can be used to retrieve co-occurrences of fish from 2 specific species. For instance, this rule can be used to analyze the interactions of juvenile Acanthurus Nigrofuscus with other species. Figure 10 shows how our user interface supports the specification of rule parameters.

2. **Manually select valid video samples**
   Users are provided with a list of video samples that satisfy the rule they defined. Users can watch the video samples. If a sample is a good example of the event of interest, users can click on the sample to include it in the training dataset. The Fig. 11 shows a selected and a discarded video sample in our user interface.

3. **Store the training dataset**
   After selecting a set of training video samples, users can label the training dataset and describe what event detection it supports. The Fig. 12 gives an example of a label for a training dataset. When storing the dataset, the system saves the rule parameters and all the video samples it retrieved: the manually selected samples, flagged as valid samples, and the discarded samples.

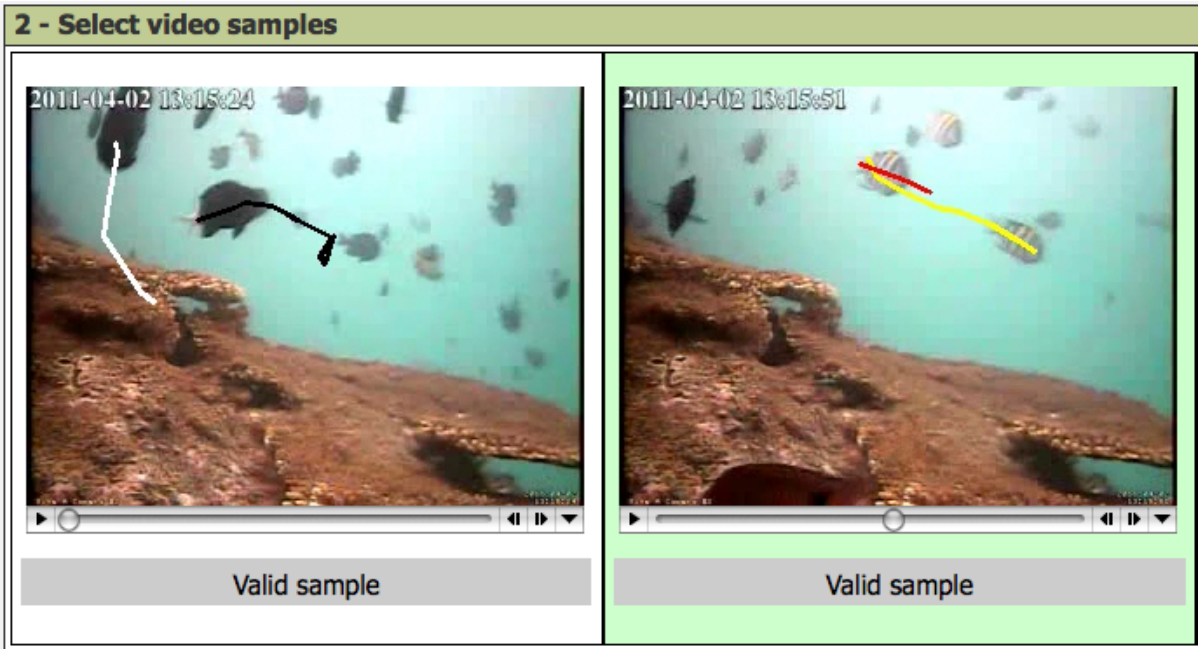


Figure 11: Users can select valid video samples (e.g., the video on the right is selected) and discard the others.

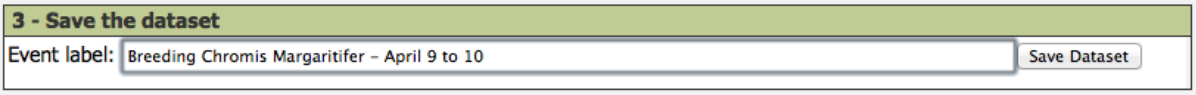


Figure 12: Users can label the training dataset to describe the targeted event.

Behaviour ID	Fish Species	Behaviour	Trajectories
DR_S	Dascyllus Reticulatus	Solitary	104
CM_S	Chromis Margaritifer	Solitary	106
PD_S	Plectrogly-Phidodon dickii	Solitary	95
PM_S	Pomacentrus Moluccenis	Solitary	60
CT_S	Chaetodon Trifascialis	Solitary	57
SB_S	Scolopsis Bilineate	Solitary	237
AC_S	Amphiprion Clarkii	Solitary	63
SF_S	Siganus Fuscescens	Solitary	51
DR_P	Dascyllus Reticulatus	Pairing	104
CM_P	Chromis Margaritifer	Pairing	144
PD_P	Plectrogly-Phidodon dickii	Pairing	138
CT_S	Chaetodon Trifascialis	Pairing	90
SB_P	Scolopsis Bilineate	Pairing	104

Table 3: Ground-truth trajectories for each fish species

4.3 Experimental Results

We proceeded to the development of an experimental component for the classification of fish trajectory. It supports the recognition of the trajectory patterns related to solitary and pairing behaviours. To evaluate our trajectory classification component, we manually annotated 13 behaviours of interest. The ground-truth datasets we produced, and the number of annotated trajectories, are described in Table 3.

We did not collect a ground-truth dataset for events described in Table 1. This is either because some behaviours were not significant for marine biologists or because we did not detect and recognize any fish of some specific species or because the number of detections was not sufficient to train our trajectory classification component.

For each event shown in Table 3, we trained a Hidden Markov Model specialized in the recognition of the trajectory patterns. Each HMM was trained using the Baum-Welch algorithm, and the number of states and output mixtures were both set to 4. For each HMM, 70% of the corresponding events were used for training and 30% for testing. In total the trajectories classification module was trained on 947 trajectories and tested on the remaining 406 trajectories.

We evaluated both A) the performance of the trajectory classification component itself, and B) the performance of the trajectory classification combined with the user-defined rules. In the case of B), the behaviour recognition system consists of applying rules that integrate i) the user-defined rule and ii) the trajectory classification. Such overall rules have the following form:

- For a given set of fish F, IF the user-defined rule for behavior B is satisfied, AND IF the trajectories of the fish F are classified as behavior B, THEN the behavior B is identified for the fish F.

Regarding A) the performance of the trajectory component alone, the Table 4 shows the classification performance of each single HMM, in terms of detection rate (DR) and false alarm rate (FAR) given in percentage. Interestingly, our HMM based trajectory classification module reached on average a DR of about 80%, and a FAR of 24%. In some cases, the number of

Behaviour ID	DR	FAR
DR_S	70.9%	35.7%
CM_S	71.8%	39.1%
PD_S	72.4%	33.3%
PM_S	100.0%	33.3%
CT_S	100.0%	33.3%
SB_S	77.4%	41.5%
AC_S	73.6%	0.0%
SF_S	100.0%	0.0%
DR_P	75.0%	27.7%
CM_P	73.9%	30.7%
PD_P	75.0%	14.2%
CT_S	71.4%	25.0%
SB_P	73.3%	33.3%
Average	**81.9%**	**24.11%**

Table 4: Performance of the trajectory classification module, evaluated alone without being combined with user-defined rules. The ***Behaviour IDs*** refer to Table 3.

false positives was relevant (e.g for DR_S about 35%), but they were then reduced when the trajectory classification was combined with the user-defined rules.

Regarding B) the performance of the trajectory classification combined with the user-defined rules, we applied the following evaluation. The performance evaluation of the event detection was assessed using normalized detection cost (NDC) (e.g., Lazarevic-McManus et al.). NDC is defined as a weighted linear combination of missed detection (MD) and false alarm (FA) probabilities. The NDC for a specific event is given by:

$$NDC = C_{MD} \cdot P_{MD} \cdot P_T + C_{FA} \cdot P_{FA} \cdot (1 - P_T) \tag{4}$$

with $P_{MD} = \frac{N_{MD}}{N_T}$, and $P_{FA} = \frac{N_{FA}}{N_T}$ that are, respectively, the missed detection and false alarm probabilities. N_E, N_T , N_{MD}, N_{FA} are, respectively, the number of the specific event instances, the total numbers of events, missed detections and false alarms. P_T is the *a priori* rate of event instances E. C_{MD} and C_{FA} are, respectively, the costs of MD and FA. We set C_{MD} and C_{FA}, respectively, to 10 and 15 to keep false alarms and missed detections balanced, as a high number of false alarms might affect fish behaviour analysis, but at the same time we do not want to miss important events.

The NDC was computed for all the species-related behavioural events of the Table 3. The results are reported in the Table 5. They highlight how our system performs quite well in detecting fish behaviour events. These results show that the system performance is comparable to those of state-of-the-art approaches performing on much simpler events.

Behaviour ID	N_E	N_T	MD	FA	P_{MD}	P_{FA}	P_T	NDC
DR_S	31	499	9	5	0.018	0.010	0.062	0.152
CM_S	32	499	9	6	0.018	0.012	0.064	0.180
PD_S	29	499	8	2	0.016	0.004	0.058	0.066
PM_S	18	499	0	1	0	0.002	0.036	0.029
CT_S	17	499	0	0	0	0	0.034	0
SB_S	71	499	16	24	0.032	0.048	0.142	0.663
AC_S	19	499	5	0	0.010	0	0.038	0.004
SF_S	15	499	0	0	0	0	0.030	0
DR_P	16	499	4	3	0.008	0.006	0.032	0.090
CM_P	23	499	6	5	0.012	0.010	0.046	0.149
PD_P	20	499	5	0	0.010	0	0.040	0.004
CT_S	14	499	4	1	0.008	0.002	0.028	0.031
SB_P	15	499	4	0	0.008	0	0.030	0.002

Table 5: Evaluation results for the behaviours recognition, performed by combing in the trajectory classification module with the user-defined rules. The *Behaviour IDs* refer to Table 3.

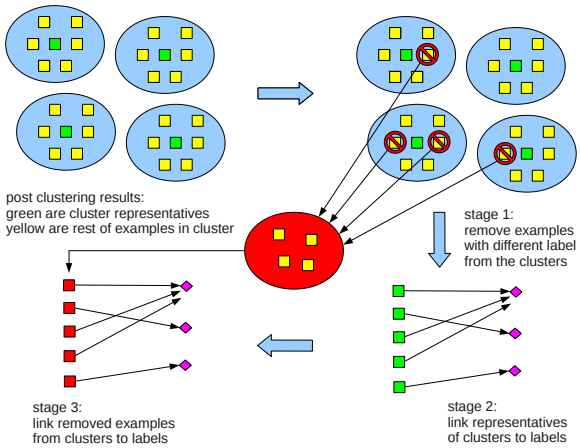


Figure 13: A schematic representation of the framework for annotating images with the support of a clustering method

5 Clustering interface for fish species recognition

5.1 Annotation Framework with Automatic Clustering

For fish species recognition, a dataset is required where given the detected fish discussed in Section 2 also the species should be labelled. However, only expert are able to provides names given an image of a fish, and because the time of experts is very limited, this does not allow us to obtain a large dataset. Another issue is that the annotation of each fish image is still very slow, in order to speed up this process we decide to use automatic methods that are able to group similar images of fish together. In this section, a framework is presented to label fish without needing much domain knowledge, where automatic methods provide a speed up in annotation allowing us to label large dataset of fish images.
The automatic methods that are able to group similar images of fish together use automatic clustering or nearest neighbor search methods (discussed in Deliverable 1.3). Most of the work presented in this chapter is already publish in [5], however some improvements will be presented that are not described before. Annotation of thousands of image can be a time consuming task, where the efficiency can be improved by using clustering or nearest neighbor search methods. A group of images (cluster or all similar images given a nearest neighbor search) is obtained using these methods together with a representative image. For the clustering method, this can be the images closest to the average image in the cluster, while for the nearest neighbor search, this is the query image. Instead of given all the images in the group a label, the user verifies if the image have the same label as the representative image. This changes the task the users from entering label names to judging if fish have similar label. This task is easier to perform than entering label name and requires less domain knowledge. The framework to annotate an entire dataset of images using a clustering method consists of three stages (Figure 13 shows a schematic of this framework):

1. Cleaning the clusters (blue ovals in Figure 13), where we remove images which are not similar to the representative image (green square).

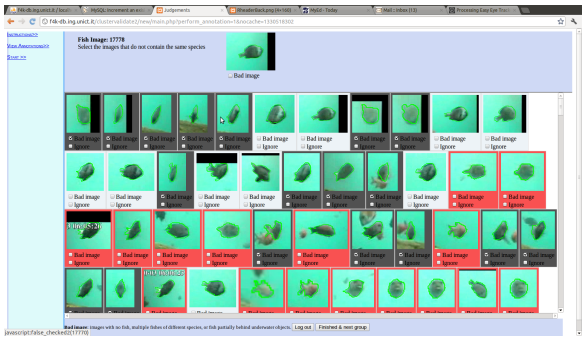
2. Merging the clusters, using the representative images of the cleaned clusters to link them to labels (shown as purple diamonds)

3. Linking removed images (shown as red squares) from the cleaning stage to the labels.

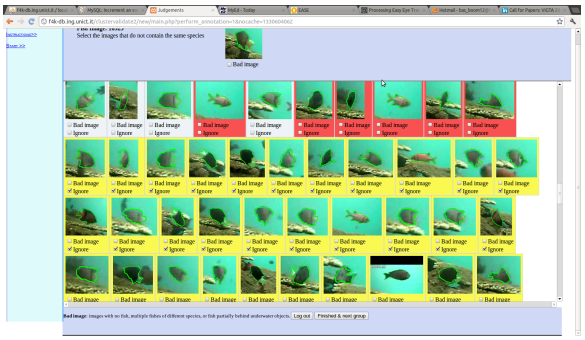
In this document, the definition for “cluster” is a group of images that are similar according to a automatic algorithm (clustering method or nearest neighbor search method). The definition for “label” is a group of images that belong to the same category according to a human annotator and furthermore contains all the images that belong in that category. In the case of the fish images, this means all the fish images that belong to a certain taxonomy.

In the fist stage, an interface for cleaning the clusters has been developed. Based on our experience in [5], some modification have been proposed in this interface to speedup the process even more.

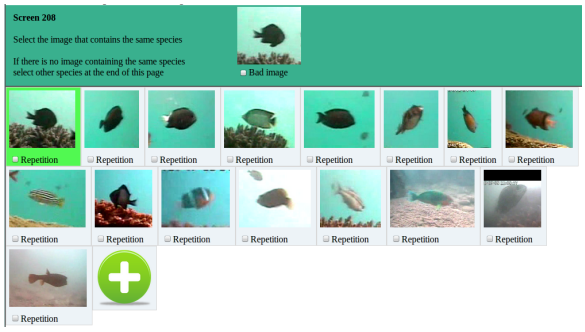
In this interface (Figure 14(a)), the fishes that do not belong to the same species as the representative image shown on top can be selected (by clicking on the image), which makes the area surrounding of the image red as shown in the screenshot. It also happens that there are images with no fish, multiple fishes of different species, fish partially behind underwater objects or uncertainty of the species because of the resolution/appearance. In this case the image can



(a) The first interface to remove images from the cluster by clicking on the image which makes the surrounding images to ignore them after a certain point making the of the image red, also "bad images" can be removed by using a checkbox making the surrounding black



(b) The first interface allows in the case of low ranked rest of the images yellow



(c) The second interface to link the representative image in the top row to a label by clicking on one of the gallery images which belonging to the same label or add a new label by pressing the green plus button

Figure 14: Interfaces

be annotate as "bad image". By checking the bad images checkbox, the area surrounding the image becomes black as shown in the screenshot. If the representative image on top is a bad image, but some of the fish images below are not, you can deselect them as bad image. It is however likely that in case of a bad representative image that there are also a lot of bad images in the cluster.

A new feature to speed up the annotation even more is that the annotator can decide to stop annotation if lots of images are not similar to the fish image shown on top. Notice that in the case of rare fish, often around 20 similar images of species exist in the database, but nearest neighbor search will find more images usually with a lower ranking. By checking the ignore checkbox, all images with a lower ranking than the current image (less similarity to the representative image) are ignored for annotation and are not saved in the database. The area surrouding these images will become yellow (see Figure 14(b).

In the second stage, the representative images will be linked to a label. Different from [5], we show this interface immediately after the interface of stage 1. Linking the representative image to a label will automatically also link the images in the cluster to the label. In this work, overclustering (e.g. 156 clusters for 32 labels) is used, and therefore there is a need to merge clusters afterwards. The second interface shown in Figure 14(c) is used to link the representative image either to a representative image of a label or a new label is created by pressing the green plus button. In the third stage, we link the set of images that are not part of a cluster, need to be linked to label as well. In [5], this is still performed for each image, however given the current dataset and the fact that we use a nearest neighbor search, the image will probably appear again and will be annotate as part of a different cluster. In the results, a fair comparison on a small subset is given based on the work performed in [5], where each removed image from the cluster is linked indiviually to one of the labels.

5.2 Combining Multiple Annotators

The previous chapter describes a strategy to annotate lots of data quickly by a single person. However often multiple persons are used for obtain image annotations and combining these annotations can be difficult. In [19] and [14], a framework for combining labels L_{ij} of image j of the M images given by user i of the N users is described. For each user i, the expertise of this user is modeled by the parameter α_i, which gives their accuracy in the annotation task. For each image j, the difficulty of the image is given by the parameter β_j. The groundtruth image label is denoted by Z_j. Expectation-Maximization on both Z_j and the parameters α_i, β_j is used to infer the final groundtruth image label given the observed label. This is extended by [14], where the label of an expert are used to first determine the parameters α_i, β_j given that the expert label a subset of images. Afterwards, the unknown remaining parameters β_j and Z_j can be compute on the entire set. In our work, the methodology of [14] is used because we have expert label available for a subset of the images.

5.3 Experiment

An experiment has been perform to compare annotation using clustering to the normal annotation. For this experiment, a dataset with 3678 automatically segmented fish images is annotated by 6 users using the KL divergence distance measure [9] between fish images, where we cluster

based on these distance using Affinity Propagation [8]. Two users annotated the dataset again using a different distance measure for clustering, basically giving us different clusters. In the case, a pyramid histogram of visual words (dense SIFT features with color information) is used and the euclidean distance between the histogram is calculated. Part of the 3678 images (159 images) is also labeled by marine biologist, where after combining the dataset of the different user and marine biologist using [14] (see previous section) 32 different fish species are found in the dataset.

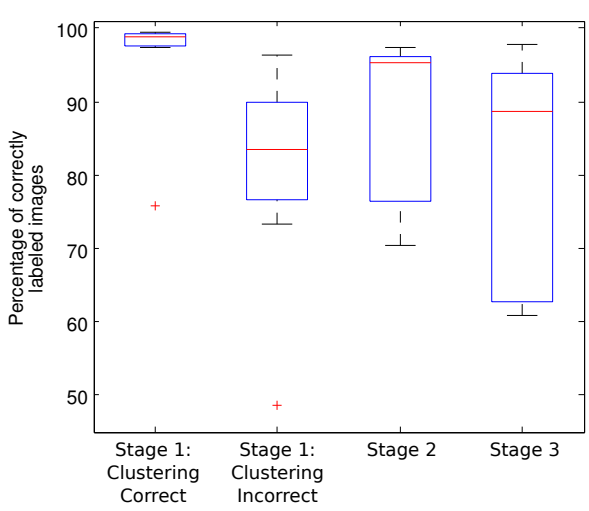
Figure 15(a) shows the accuracy at each of the stages. Because the annotation in the first stage depends on the clustering performance, this stage is divided into two boxplots. It is clear from these boxplots that users make more mistakes with removing the incorrectly clustered images than with correctly clustered images. We assume that this has two causes: The first cause is that users do not scan the images very comprehensively, which leads to labeling mistakes which could be avoided. The second cause is that some images are hard to recognize and users might not be able to separate them correctly. The performance in stage 2 (see Figure 15(a)) is a good indication of the labeling performance without using clustering, because stage 2 has the user select a pictorial "label" for each presented image. In our case, we only present the representative images rather than the full set, but we argue that accuracy would be similar if all images were presented. From the performance of stage 3, we observe that it is also more difficult to link the images excluded from stage 1 (which were incorrectly clustered), than linking the representative images.

5.3.1 Accuracy of Annotations

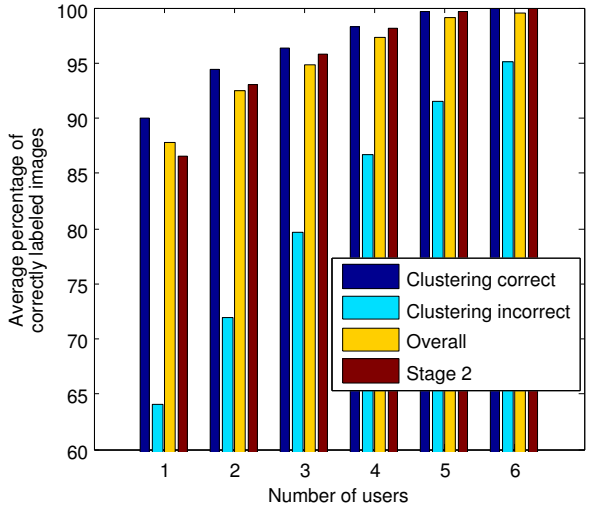
In order to obtain the accuracy of the annotations, we first obtain a groundtruth dataset based on all the annotation of both users and biologist. Given this dataset, we compare how all combinations of different number of users who labeled with the KL divergence give different label compared with this dataset. The results of these experiment are shown in Figure 15(b), which gives the average performance in annotation given a all combination of a certain number of annotators. The "Overall" results show the accuracy of annotation with clustering, while the "Stage 2" results give an estimation of the accuray of annotating the images without a clusting methods. In Figure 15(b), we show a small decrease in accuracy if clustering is used to support the annotation. The first bin of Figure 15(b) shows the user performance of correctly clustered images while the second bin shows the performance on incorrectly clustered image. Although much more mistakes are made on the incorrectly cluster images, their influence on the entire system is smaller because the percentage of incorrectly clustered images for KL divergence and Pyramid histograms is respectively 9.8% and 16.9%.

5.3.2 Improvements in time and mouse clicks:

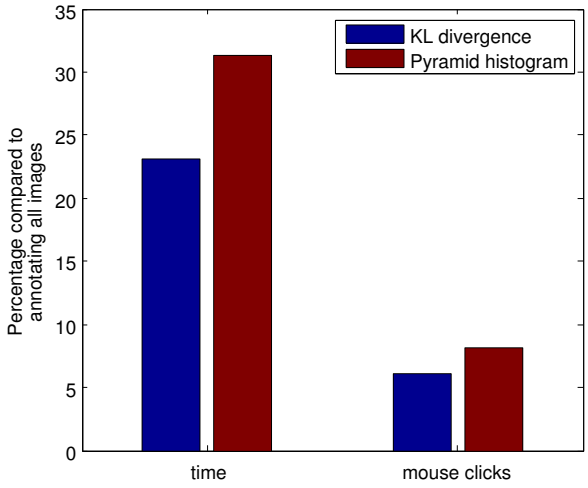
In Figure 15(c), we show the improvement in time and mouse clicks. To estimate the time, one user performed non-stop annotations for us while we measure the time it took to finish one screen. This allowed us to measure that the average time the complete the first interface is 19.7 seconds while the average time to complete the second interface is 7.3 seconds. In Figure 15(c),



(a) Box plot of the performance in labeling of all users for the individual stages



(b) Histogram of the average combined performance of a certain number of users



(c) The improvement in both time and mouse clicks over annotating all images

Figure 15: Evaluation

we extrapolated these values for all users and both clustering methods and compared them with only using the second interface for all images. We also measured the number of mouse click, which can be important in crowdsourcing because users get paid by the amount of clicks. By labeling all M images with the second interface, $2M$ clicks are necessary (where one click allows the user to select the correct fish and one click is necessary to confirm the selection). By using clustering, we only need to click on a small amount of images that does not belong to the cluster and for each cluster an extra click is necessary to confirm your selection again. Afterwards, only for the representative images we have to perform the second interface. This results in a reduction of 93% in mouse clicks when using the KL divergence.

5.4 Conclusions

Although there is a very small decrease in the accuracy compared to not using clustering to annotate these images, there is a major improvement in time and mouse clicks. It takes users a third of the time to annotate with clustering support compared to annotating the dataset without clustering support. This also means that three users with clustering support can annotate the data to achieved better accuracy in the same time as one user without clustering support. The accuracy of these three users according to Figure 15(b) is much better than the accuracy of one user. These difference in quality also get smaller if more users are annotating. Currently, we have decided that the accuracy achieved with 3 users is good enough for us at the moment. This framework has also been used without stage 3 to label a dataset of around 23000 fish images, which took about 8 hours for each of the three users annotating. Currently, we have annotated around 91894 images where a lot of images are labelled as "bad images", because of occlusions, low resolution, etc. For 28,264 images, we have obtained species label at the moment, however these numbers are still increasing. New efforts are being focussed on finding rare species in this data, ignoring the common species where enough training and testing data is available.

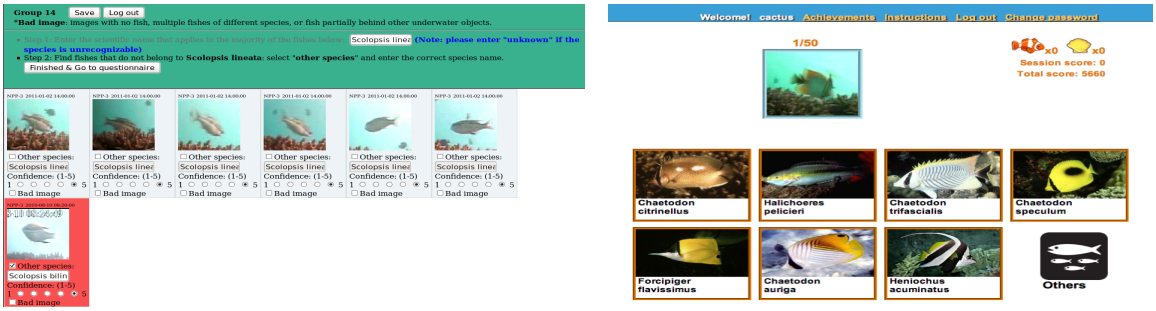
6 Validate annotation quality in fish species recognition

To be able to validate the label qualities for fish recognition task, we need expert to verify the labels collected in the previous stage. Experts are, however, expensive and a scarce resource, e.g. biologists specialized on the Australian reefs perform not as good as those specialized on the fishes that live on the Taiwanese reefs. Therefore we use their expertise to transform the difficult fish labeling task into a game based on a visual similarity comparison task that can be performed by large numbers of non-experts. In the game, players are shown a single *query image* along with multiple labeled images of candidate species, referred to as *candidate labels*, and are asked to assign the query image to the label that depicts the same species as the fish in the image.

6.1 Annotation with experts

We ask our experts to label only a small subset of our data and developed a cluster-based interface to facilitate their labeling process. The images labeled by the experts are used as gold standard for the evaluation of non-expert labels.

We manually clustered 3000 images randomly chosen from our video data. We present the images to the experts in a labeling interface as shown in Figure 16(a). In total 28 clusters were obtained. Using this interface, the expert first enters the species name that applies to the majority of the images in a cluster. Once the name is entered, all images within the cluster are automatically assigned with the same species name. Then, the expert is asked to select those images that should not belong to that cluster. By selecting these images, he/she can also input the correct species names for them. In this manner, in the worst case, the expert will have to manually assign a species name to each of the images, i.e., when the clustering is so bad that each image within a cluster represents a different fish species. In the best case, i.e., when the cluster is pure, the expert only needs to enter the species name once. After finishing annotation, we submit the expert to a questionnaire in order to collect information such as whether the labeling task was difficult for him/her, and why it was difficult. To limit the amount of effort



(a) Species recognition interface for experts (b) Game interface for non-experts

Figure 16: Expert and game interfaces for labeling fish species.

experts need to examine the clusters, at most 30 images are randomly selected from each cluster and shown to the experts. As the size of the clusters is unevenly distributed, e.g., only some of the clusters contain more than 30 images, we obtain a total of 190 labeled images.

We invited 3 marine biologists (referred to as E1, E2 and E3) to participate in the expert labeling task. They have research experience of 30, 10 and 25 years in Taiwanse coral reef fish, respectively.

We use Cohen's kappa [6] to measure the agreement between the expert labels, assuming the complete category set consists of all unique species mentioned in the labels provided by the experts.

	Species level		Family level	
Comparison	Avg.κ	Sdv.	Avg.κ	Stv.
E1 vs. E2	0.55	0.008	0.85	0.004
E1 vs. E3	0.48	0.008	0.75	0.000
E2 vs. E3	0.67	0.006	0.76	0.0001

Table 6: Cohen's kappa for measuring expert agreement.

We find that biologists do not always agree with each other. Further, sometimes the biologists are not sure which species a fish should belong to: 1) one of the experts assign labels such as "A or B" to 3 images, and 2) in 45 cases (each case is a pair of image and expert, in total we have 190 x 3 cases) a family or higher level label is assigned. In the former case, we consider both labels mentioned, and in the latter case, we consider all species under a higher level label as possible target labels. Thus it is possible that an image has multiple labels assigned by a single expert. In total, 288 species and 20 families were mentioned as labels for the 190 images. When there exist multiple labels for an image assigned by one expert, we randomly draw one of them as the target label being evaluated; we repeat this process 100 times and report the averaged κ and its standard deviation over the 100 runs[3]. We evaluate labels at both species and family level.

Results in Table 6 show that at the species level, the agreement between experts is only moderate, while at family level, a much stronger agreement can be found, but still not perfect. This result suggests that our labeling task is not trivial even for experts. Further, from the questionnaire we learn that according to the experts, the top factors that make recognition

[3]Notice that the agreement calculated in this way is rather conservative

difficult are: 1) the low quality of the images; and 2) the fact that some species are visually very similar and not distinguishable using the features that can be observed from the images. For example, the main feature biologists use to distinguish Chromis Chrysura and Chromis Margaritifer is their body size, while in video footage the size of a fish depends on its distance to the camera, and therefore it does not provide enough information.

6.2 Annotation with non-experts

6.2.1 Interface

For non-experts, a labeling game interface as shown in Figure 16(b) is used. The players are asked to compare a *query image* (i.e., the image to be labeled) to a set of *candidate labels*, i.e., labeled images of candidate species. To avoid overloading the players with too many candidates, we limit the number of candidates to 7. They choose from one of the candidates if they believe that the fish in that candidate label and the query image belong to the same species, or "others", if none of the candidates is similar enough to be considered as the correct answer. The labeling task is then a multiple choice based on the perceived visual similarity between the query image and the candidate label images. The player receives a feedback score for each choice he/she makes. Ideally, he/she can learn from the feedback and tries to improve his/her performance.

We divide the labeling process into sessions of 50 query images. This gives a break as well as a goal for the players. It typically takes 5 to 10 minutes to complete a session, depending on whether the player is familiar with the system and the task.

In order to increase user engagement with the labeling task, we included competition elements. We show the top 10 scorers (those who have achieved top scores in single sessions), and top contributors (those who have achieved accumulative top scores), which is meant to encourage people to achieve higher scores and play more sessions.

6.2.2 Experiment setup

From the data obtained from the experts, we find that 53 out of the 190 were assigned to "wrong" clusters during the manual clustering stage. That is, there exist many fish that look similar but do not belong to the same species. We thus question: *Can non-experts distinguish between similar species when examples of these species are displayed next to each other?* To find answers to these questions, we conduct two experiments that simulates two situations.

Experiment 1 We assume an ideal situation, where the *target label(s)*, i.e., labels suggested by the biologists, of the query image is always among the candidates. The primary goal of the experiment is to investigate whether the players can identify the target label when there exist very similar species.

We select candidates that are similar to the target labels as follows. Recall that we have manually created clusters. These clusters are not always pure according to experts' labels. We find that 53 out of the 190 were assigned to "wrong" clusters during the manual clustering stage. Let $c = \{i_n\}_{n=1}^{N}$ be a cluster containing N query images, and $f(i)$ maps an image to one of the species $S = \{s_m\}_{m=1}^{M}$. We compute a relevance score between an image $i \in c$ and a species as $\text{score}(i, s) = \text{count}(f(c) = s)/N$. All species with a non-zero score are the ones that were clustered together, which means that they are visually similar. We select top 7 species

as candidates. If less than 7 species were available, we fill the remaining slots with random images. If more than 7 species have non-zero scores, e.g., in case the biologists have assigned multiple labels to some of the images in the cluster, we make sure that the target labels are in the candidates.

Experiment 2 We then consider a more realistic situation when some target labels are not in the candidates. Notice the number of candidates is way smaller than the number of all possible fish species, e.g., 288 if we consider all the species mentioned by the biologists for the 190 images. In practice, we do not have information about the target labels of the query images. We need to select candidates based on certain similarity measures computed with automatic methods, which are most likely not perfect. It is then important to know whether the non-expert players can still make right choice, i.e., select "others" when similar species are displayed as candidates.

We use the same setting as in Expr.1 to select candidates and deliberately remove the target labels from the candidates for a set of randomly selected query images. On the one hand, we want sufficient cases where the target labels are removed, on the other hand, if too many target labels are removed, the users may expect that "others" is always the safe bet when they are not sure. With a few trials, we decide to remove the target labels for 25% of the query images.

6.2.3 Settings of feedback scores

In this study, we only consider the simplest case for the system feedback, i.e., feedback from the expert labels. Specifically, we assign scores to each option of the candidates based on the biologists' voting. Since experts do not always agree, a click on an option can receive 0, 1, 2, or 3 points, depending on the number of biologists agree that it is the target label. In practice when expert labels are not available, of course, other types of feedback should be used, e.g., peer-agreement, automatic similarity measures. We leave questions such as how these feedbacks influence the user learning behavior to future investigation.

6.2.4 Data obtained

We use convenience sampling to collect our players. We launched our game in our own social network as well as in public events, e.g., demo exhibitions. Our users have a diverse background and age groups, including school age children as well as university students, researchers, etc. We collect labels for the 190 images that are labeled by the experts. 22 players contributed 72 sessions in Expr. 1 and 32 players contributed 49 sessions in Expr. 2. On average each image received 19 and 13 labels, respectively. Notice that in Expr. 2 we have more players but less sessions. This is because most of the sessions of Expr. 2 were done in a public event, where people typically try out for just one session. Four players have participated both experiments and in total played 9 sessions in Expr. 2. In our evaluation of Expr. 2, we will treat their contributions separately, as they may have been trained in Expr. 1 and their performance is not comparable with those who were new to the game.

6.2.5 Aggregating non-expert labels

Since each query image is associated with multiple labels from multiple players, we need to aggregate them into a single assignment for evaluation. We consider two simple strategies.

	Species						Family					
	E1		E2		E3		E1		E2		E3	
	Avg. κ	Sdv.	Avg. κ	Sdv.	Avg. κ	Sdv.	Avg. κ	Sdv.	Avg. κ	Sdv.	Avg. κ	Sdv.
U.random	0.51	0.03	0.53	0.03	0.45	0.03	0.73	0.03	0.71	0.03	0.62	0.03
U.MVote	0.62	0.01	0.65	0.006	0.55	0.009	0.83	0.008	0.81	0.01	0.72	0.009

Table 7: Agreement (Cohen's kappa) between experts and non-experts (correct labels present).

With the first strategy, we randomly select one of the players' labels as the chosen label for the query image. If we run this random aggregation, say, 100 times, we obtain a sample of 100 label assignments given random labels from random players. By evaluating the result of the randomly aggregated labels, we obtain an expected performance of a single player. The second aggregation strategy is majority voting. Since experts may give multiple labels to an image, we do not simply take the winner of the majority voting as the chosen label, but rank the candidates in descending order of their votes.

In Expr. 2 when target labels are not displayed, the labels "others" can be correct but not providing information about which label should be assigned to the image. We ignore these labels when aggregating as they neither hurt or help the performance.

6.3 Evaluation

We use two measures to evaluate non-experts' performance. (i) One natural way to evaluate the non-expert performance is to measure the agreement between the non-expert labels and the expert labels. Again, we use Cohen's kappa [6]. We have already seen that the marine biologists often disagree on the species names among themselves. If the experts cannot agree on their labels, it is probably unreasonable to require the non-experts achieve an extremely high agreement with the experts. (ii) While the agreement analysis provides us with insights of the alignment between non-experts and experts, it does not provide an intuitive indication of how correct the obtained labels are. Further, we do not have a principled way to handle the multi-label situation with κ. We therefore also evaluate using of NDCG [11], which handles multi-labels and provides a more intuitive interpretation of the correctness of the labels. For a query image, given the biologists' judgment, each candidate can be rated as 0, 1, 2, or 3. The ranked list of candidates generated by the (majority) voting aggregation is then evaluated using this grated expert judgements.

6.4 Results

6.4.1 Performance when target labels are present

Table 7 shows the result of label agreement at both species level and family level. If we compare Table 7 to Table 6, we see that even with a random aggregation, the agreement between expert and non-expert labels are rather similar to that among the experts themselves. Recall that the κ values of experts agreement ranges from 0.48 to 0.67 at the species level and from 0.75 to 0.85 at the family level. The result of majority voting has a stronger agreement with the experts compared to the random aggregation results. This indicates that the crowd can, to some extent, correct errors made by individuals. Further, Table 8 shows the performance of non-expert labels in terms of NDCG. In practice, when using the collected labels as training data, often only the

label(s) with the highest scores will be considered as target labels. Therefore it is important that the very top ranked labels are the correct ones according to experts' labels, we list the results of NDCG@1 and 5. Unlike the agreement comparison, here we do not have a baseline to compare to. However, we do see that the scores at least indicate that for a majority of the images, the non-experts have made correct choices. Since random aggregation only has one chosen label, below NDCG@1, no further gain can be achieved. For majority voting, we see that some other relevant candidates are within the top 5 of the ranked list, as the scores at NDCG@5 are higher than that at NDCG@1. That is, when present among the candidates, in most cases the target

Method	Species		Family	
	NDCG@1	NDCG@5	NDCG@1	NDCG@5
U.random	0.71	0.67	0.85	0.82
U.MVote	0.84	0.88	0.93	0.94

Table 8: Non-experts' performance evaluated by NDCG (correct labels present).

labels can be identified by the players. In particular, the agreement achieved between the non-experts and the experts are comparable to that achieved among the experts themselves.

6.4.2 Performance when some target labels are absent

Intuitively, this is a more difficult task for the players, as we deliberately removed the target labels of some of the query images, and left in candidates that are similar but not actually relevant.

Table 9 shows the agreement between the non-experts and the experts. The "new" players are those who only participated in Expr. 2, while "old" players participated in both experiments. We see that for new players, the randomly aggregated labels have a much lower agreement with the experts compared to those in Table 7. However, the results after majority voting are much better. This suggests that the crowd can help to correct some of the errors made by new individual players. On the other hand, "old" players perform comparable to the results in Table 7. Since we only have 4 old players, random aggregation is not very different from majority voting.

Table 10 lists the results in terms of NDCG. The new players in Expr. 2 have a significant lower performance compared to Expr. 1, both with random aggregation and majority voting. In general the new players in Expr. 2 perform worse compared to Expr. 1. We consider two potential explanations: 1) the set up of Expr. 2 makes a more difficult task for novice players; or 2) since most of the new players did only one session, the general quality of the labels are not as good as that of Expr. 1, where many played more than one session. To distinguish the two cases, we verify if the results from only the first session of each player in Expr. 1 still outperform that of Expr. 2. In Table 11 we see that indeed, there is a significant difference between the performance of the first session labels in the two experiments. That suggests that when target labels are absent while similar non-target labels are present, the novice players are more likely to be confused. This confirms our intuition that selecting a good set of candidate labels is very important.

User	Method	Species						Family					
		E1		E2		E3		E1		E2		E3	
		Avg. κ	Sdv.	Avg. κ	Sdv.	Avg. κ	Sdv.	Avg. κ	Sdv.	Avg. κ	Sdv.	Avg. κ	Sdv.
New	U.random	0.47	0.03	0.37	0.03	0.36	0.03	0.60	0.03	0.59	0.04	0.58	0.04
	U.MVote	0.65	0.009	0.50	0.008	0.45	0.009	0.73	0.01	0.73	0.01	0.68	0.01
Old	U.random	0.52	0.02	0.67	0.02	0.62	0.02	0.79	0.02	0.77	0.02	0.71	0.02
	U.MVote	0.53	0.01	0.68	0.01	0.64	0.02	0.80	0.02	0.78	0.02	0.74	0.01

Table 9: Agreement (Cohen's kappa) between experts and non-experts (some labels missing)

Method	Users	Species		Family	
		NDCG@1	NDCG@5	NDCG@1	NDCG@5
U.Random	T1	0.71	0.67	0.85	0.82
	T2.new	0.52$^{\blacktriangledown}$	0.50$^{\blacktriangledown}$	0.66$^{\blacktriangledown}$	0.68$^{\blacktriangledown}$
	T2.old	0.86$^{\blacktriangle}$	0.81$^{\blacktriangle}$	0.91$^{\blacktriangle}$	0.91$^{\blacktriangle}$
U.MVote	T1	0.84	0.88	0.93	0.94
	T2.new	0.72$^{\blacktriangledown}$	0.77$^{\blacktriangledown}$	0.86$^{\blacktriangledown}$	0.94
	T2.old	0.88	0.86	0.91	0.94

Table 10: Comparing the performance of players under Expr. 1 (T1) and 2 (T2) ("new" and "old" players). .new refers to new players; .old refers to old players. of players under the settings of experiment 2 to that under the settings of experiment1, in terms of NDCG. T1 refers to experiment 1; T2.new refers to experiment 2 with new players; and T2.old refers experiment 2 with old players from experiment 1. $^{\blacktriangle}(^{\blacktriangledown})$ indicates a significant (p-value$<$0.01) difference tested using Wilcoxon signed-rank test [20].

Method	Users	Species		Family	
		NDCG@1	NDCG@5	NDCG@1	NDCG@5
U.Random	T1	0.72	0.67	0.84	0.82
	T2	0.60	0.57	0.76$^{\blacktriangledown}$	0.77$^{\blacktriangledown}$
U.MVote	T1	0.84	0.88	0.93	0.94
	T2	0.72$^{\blacktriangledown}$	0.77$^{\blacktriangledown}$	0.86$^{\blacktriangledown}$	0.94

Table 11: Comparing the performance in the first sessions under Expr. 1 (T1) and 2 (T2). In T2, only "new" players are considered. Wilcoxon signed-rank test is used for significance testing.

6.5 Conclusion

We converted an image labeling task that requires extensive domain knowledge into an image matching game that is based on visual similarity comparison only. When the correct labels are always presented among the candidate labels, non-experts can play this game rather well: domain experts agree as often with the aggregated game labels as they agree with each other's

labels. When the game is played under the more realistic condition that the correct label is not always presented, performance of novice users drops, but players that had played the game before still performed as good as under the ideal condition.

A number of directions are left to be explored in the future. We used feedback from the experts, while in practice, the game will rely on automatic feedback or peer-agreement. The influence of feedback quality on users' performance and learning behavior is yet to be studied. Similarly, components within our labeling system such as the selection of candidates in practice will have to rely on automatic methods. While our user study have provided insights into how these components influence user performance, it remains unexplored how these should be integrated as a full fledged interactive system. Finally, we need to investigate how our approach can be extended to other domains such as medical image annotation.

References

[1] Luis von Ahn. Games with a purpose. *Computer*, 39(6):92–94, June 2006.

[2] Amol Ambardekar, Mircea Nicolescu, and Sergiu Dascalu. Ground truth verification tool (GTVT) for video surveillance systems. In *Proceedings of the 2009 Second International Conferences on Advances in Computer-Human Interactions*, ACHI '09, pages 354–359, 2009.

[3] Marco Bertini, Alberto Del Bimbo, and Carlo Torniai. Automatic video annotation using ontologies extended with visual information. In *Proceedings of the 13th annual ACM international conference on Multimedia*, MULTIMEDIA '05, pages 395–398, 2005.

[4] Lukas Biewald. Massive multiplayer human computation for fun, money, and survival. In *Proceedings of the 11th international conference on Current Trends in Web Engineering*, ICWE'11, pages 171–176, 2012.

[5] Bastiaan J. Boom, Phoenix X. Huang, Jiyin He, and Robert B. Fisher. Supporting ground-truth annotation of image datasets using clustering. ICPR 2012, to appear.

[6] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, April 1960.

[7] D. Doerman and D. Mihalcik. Tools and techniques for video performance evaluation. In *Proceedings of 15th International Conference on Pattern Recognition.*, volume 4, pages 167–170, 2000.

[8] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972 –976, February 2007.

[9] J. Goldberger, S. Gordon, and H. Greenspan. Unsupervised image-set clustering using an information theoretic framework. *IEEE Trans. on Image Processing*, 15(2):449–458, February 2006.

[10] P. Heroux, E. Barbu, S. Adam, and E. Trupin. Automatic ground-truth generation for document image analysis and understanding. In *Proceedings of the Ninth International*

Conference on Document Analysis and Recognition - Volume 01, ICDAR '07, pages 476–480, 2007.

[11] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002.

[12] C. Jaynes, S. Webb, R.M. Steele, and Q. Xiong. An open development environment for evaluation of video surveillance systems. In *PETS02*, pages 32–39, 2002.

[13] I. Kavasidis, S. Palazzo, R. Di Salvo, D. Giordano, and C. Spampinato. A semi-automatic tool for detection and tracking ground truth generation in videos. In *VIGTA '12: Proceedings of the 1st International Workshop on Visual Interfaces for Ground Truth Collection in Computer Vision Applications*, pages 1–5, New York, NY, USA, 2012. ACM.

[14] Faiza Khan Khattak and Ansaf Salleb-Aouissi. Quality control of crowd labeling through expert evaluation. In *Second Workshop on Computational Social Science and the Wisdom of Crowds (NIPS 2011)*, pages 1–5, Dec 2011.

[15] Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. Collecting image annotations using amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 139–147, 2010.

[16] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vision*, 77(1-3):157–173, May 2008.

[17] Alexander Sorokin and David Forsyth. Utility data annotation with Amazon Mechanical Turk. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, June 2008.

[18] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, pages 319–326, New York, NY, USA, 2004. ACM.

[19] Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Jacob Bergsma, and Javier R. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, pages 2035–2043, 2009.

[20] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[21] Jenny Yuen, Bryan C. Russell, Ce Liu, and Antonio Torralba. Labelme video: Building a video database with human annotations. In *ICCV'09*, pages 1451–1458, 2009.

