View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Archivio istituzionale della ricerca - Politecnico di Milano

Designing Bots in Games with a Purpose

Giorgia Baroffio, Luca Galli, Piero Fraternali
Dipartimento di Elettronica, Informazione e Bioingegneria, Via Ponzio 34/5, 20133 Milano, Italy
Email: name.surname@polimi.it

***Abstract*—The massive amount of time that people spend in online gaming is being increasingly exploited by a particular kind of Serious Games, the Games with a Purpose (GWAP), used to solve complex problems as a byproduct of a collaborative gameplay. The required tasks are solved by exploiting game mechanics that often require the submission of thousands of players' annotations, to achieve a robust estimate of the results. Gathering a consistent playerbase able to solve computational problems at a scale is extremely difficult, due to better entertainment alternatives on the market and the necessity of pairing each player with another one due to the inherent multiplayer nature of this genre. Artificial players (bots) may be introduced when the online platform has not enough human contributors to employ, but their functional requirements and implementation is much different than the one of traditional videogames. In this work we describe the framework and the design choices that have been used to implement a bot in an existing GWAP for fashion garment image segmentation, showing how supervised methods can be applied effectively to emulate human behaviour in the resolution of computational tasks through gameplay actions.**

I. Introduction

Game technologies, terminologies and practices are transcending the boundaries of pure entertainment. This phenomenon can be appreciated by the growth of a special subset of serious games, also known as Games with a Purpose (GWAP), as an industry and research field. A Game with a Purpose is a game in which players generate useful data as a by-product of play [1]. Despite the success of games like ESP [4] or FoldIt [10] and the considerable number of GWAP that has been developed, not all of them have been able to attract the stable playerbase necessary to solve the tasks for which they have been created. This is a severe limitation since the effectiveness of the game mechanics of a typical GWAP rely on the presence of two or more players per game session; paired gameplay sessions allows players to validate each other's computation.

If not enough players are online to start a round and if it is difficult to guarantee a stable presence of active users over time, a GWAP is destined to failure. The system may compensate for these situations by automatically enabling Artificial Players (bots) to play with logged in players. Although a lot of effort is put in Computational Intelligence research for traditional videogames, few publications discuss the topic in the context of GWAP.

In this paper we describe how we have solved the problem of creating the single-player mode for a novel GWAP, *Sketchness* [12], implementing ad-hoc bot players able to fulfil asymmetric roles that could not be emulated just with traditional techniques used in the field (e.g. replay of human submitted traces). The artificial players combine the use of the best traces, submitted by the players during the game, with supervised learning techniques to build a model for the recognition of garments in a collection of fashion images. This goal is achieved using as hint just the contour of the object hand traced by a player.

The proposed approach is validated with an experimental analysis that makes use of seven different supervised learning techniques, i.e. Decision Trees, Naive Bayes, Random Forest, Random Trees, Neural Networks, Support Vector Machines and Bayes Networks. The feasibility of the implemented bot is finally checked during a "Turing Like" test involving thirty players in a controlled setting.

The paper is organized as follows. In Section II we provide a brief overview of the related works in the literature. We introduce the design and implementation choices of the GWAP subject of this work in Section III. In Section IV we outline the approach followed to implement artificial players within the game and in Section V and VI we report the experimental results and tests with the target users. Finally, we draw our conclusions in Section VII.

II. Related Work

Given the recent nature of the genre, academic research on the design and implementation of AI in GWAP is still extremely limited. Most works on GWAP focus on embedding a specific problem solving task into an enjoyable user experience and on evaluating the quality and quantity of output produced by players.

For tasks too expensive to be addressed with state-of-the-art computer algorithms, crowd wisdom has been used to quickly search and reduce the space of possible solutions, as it happens with the FoldIt game [9], in which crowds of online users compete and cooperate to predict biologically relevant low-energy protein conformations in the form of a 3D puzzle.

The classification of alternative game design patterns, based on different input-output templates, discussed in [2], is the first attempt to generalize GWAP design principles, while a comprehensive list of the games that have been developed so far is provided in [16].

Output agreement games induce humans to pro-

duce semantic annotations that describe as accurately as possible the input and to obtain a match for tasks like image tagging [4], image preference elicitation [13], ontology construction [25] and sentiment analysis [24]. In *Input Agreement* games, players must understand if they have been provided with the same content by describing it to the other players. Both input and output agreement games assign equivalent roles to the players, whereas the *Inversion-problem* template differentiates between them: at each round, one player assumes the role of the "describer"and the other one the role of the "guesser". The describer receives an input (e.g, an image, or a word) and based on it, sends suggestions to the guesser to help her identify some feature of the original input. Peekaboom [1] uses this approach with the aim of detecting objects in images.

In the case of an input-agreement game or output agreement game, such as the ESP game [4], implementing an Artificial Intelligence (AI) is relatively an easy task, accomplished by replaying prerecorded actions submitted by the players in previous gameplay sessions. In inversion-problem games, implementing bots is much harder because one of the players, the guesser, must dynamically react to human players' actions; just few GWAP like Peekaboom [1], Phetch [3], and Verbosity [5] have artificial players often skewed towards the easier role to be emulated or able to solve just basic tasks.

III. Sketchness

Sketchness [12] is a multiplayer Game with a Purpose used to obtain segmentations on fashion related images that couldn't be processed automatically by state of the art garment segmentation algorithms [26] [27]. It takes inspiration from *draw and guess* games like the famous boardgame Pictionary[1]. The players take turns in acting like "Sketchers" and draw the shapes of objects in a provided image, in order to make the other players, the "Guessers", guess the underlying object using as a hint just the traces drawn by the first player.

If the icon related to the right garment is chosen, both the drawer and the players that were able to spot the object will receive points based on the time the response was submitted. After a certain number of rounds, in which each player is asked to draw on different images, the winner will be the player who has achieved the highest score. In each round a player is chosen at random to be the Sketcher, as in Figure 1 while all the others will play as Guessers, as in Figure 2.

The game can be used to: 1) Check if a particular fashion item is present or not within an image by asking for a confirmation to the crowd in the form of a tag (a textual annotation identifying an object present in a picture); the image can also be tagged in the case in which it was not previously annotated. 2) Segment the tagged fashion item within the image by asking to the players to trace the contours of the object. To increase the enjoyability on mobile devices and make the game language agnostic, textual tags are represented with icons depicting garments.

Fig. 1: Sketcher's Interface

Fig. 2: Guesser's Interface

Figure 3 presents the architecture of the implemented game and the associated content management system used to store images and associated metadata. Being a GWAP, Sketchness has to be played online by as many players as possible, without requiring any particular software or equipment other than the ones commonly used to browse the Internet. For these reasons, the game has been developed in HTML5 following the MVC software architecture pattern, using the Play! Framework[2] for the backend and a custom Content Management System (CMS) developed in NodeJs[3] to store the annotations submitted by the users.

The game itself has been developed as a complex state machine created in a specular way both

[1]https://boardgamegeek.com/boardgame/2281/pictionary

[2]https://www.playframework.com/

[3]https://nodejs.org/

in the backend (Java) and in the frontend (HTML5 + Javascript). The messages among the server and the clients are exchanged by using specialized JSON packets sent via websockets, ensuring real time responsiveness. The views represent the interfaces of the game that are in charge of emitting events related to the gameplay actions performed by the players.

The controller is in charge of monitoring and filtering only the meaningful received events, forward them to the models and display the updated status of the system through the views. The models are used to handle the gameplay of the game and persist the operations performed by the player with the use of a custom CMS. A message bus allows the game to be divided in independent modules able to communicate among each other, turned off or reused in future applications.

The GameRoom contains the logic of the game that assigns the roles for the players in each round, associate segmentation tasks to be solved, verifies the answers of the players and assigns points following the rules defined in the finite state machine that controls the logic; it is the only module that requires modifications to be able to build a different game.

The other independent modules are the Chat, that contains the logic for the creation of a chatroom used to exchange messages among the players and the game and the Paint module, which allows the creation of a canvas that can be used to share drawings and images. The CMS module receives packets from the shared message bus and provides interfaces and methods to handle the storage of actions and annotations for future usage, in particular to calculate the reputation of users and store the aggregated masks for a particular garment.

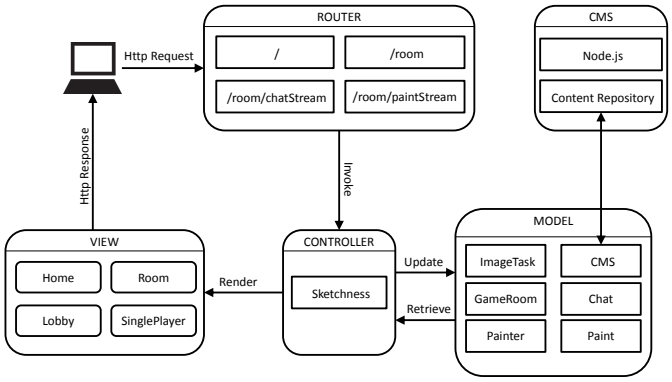

Fig. 3: Architecture of the Sketchness GWAP

IV. Bot Implementation

In the following, we propose an approach to develop an artificial player able to take into account (i) the particular annotation task to be solved and (ii) the current game state and contributions of the human player. At first, we have collected annotations for all the images stored in the CMS that stores all the actions submitted in the game from real gameplay sessions with humans. Then, we preprocess the collected game data and the metadata associated to the images to generate a suitable dataset for applying supervised classification methods to learn one or more models of the target garment identification strategy. At the end, the learned models are deployed to the actual game engine to replace the actions and decisions made by a human player.

Field	Description
id: Numeric	a unique identifier for the action
image: Image	the image on which the action has been performed
tag: String	the name of the garment to segment
user: Numeric	the ID of the player within the backend storage
type: String	the action type (tagging, segmentation)
segmentation: Segmentation	the segmentation metadata for segmentation actions
started_at: Date	the date at which the action started
completed_at: Date	the date at which the action finished

TABLE I: Description of the attributes associated to a gameplay action

A. GWAP Bot Roles

In the GWAP there are two possible roles that an artificial player may perform:

1) Sketcher: Sketchers are provided with images coming from a collection of (not)annotated fashion items. They are the only ones with the rights to see the image during a round and need to provide a tag for a garment visible in the image, such as “tie” or “trousers” or are given a tag generated from previous matches. Once the tag has been added, the Sketcher is asked to draw the selected object by tracing its contour. An example of a Sketcher’s interface has been provided in Figure 1.

2) Guesser: Guessers are able to see just the contour traced by the sketcher and are asked to pick their guesses by choosing among nearly 30 icons depicting different clothes in order to match the garment’s tag; the quicker the player answers correctly, the higher the score. An example of a Guesser’s interface has been provided in Figure 2.

B. Logging Game Data

To model players’ specific behaviours of the two aforementioned roles, the first step involved the acquisition of a meaningful set of traces submitted by humans during the game in the form of gameplay actions.

An *Action* object describes an action that a user can perform within a gameplay session; its fields are described in Table I.

A *Segmentation* is a type of action performed by a user that stores the points used to reconstruct the path drawn by the Sketcher during a round.

The segmentation field is used to store the points associated to the segmentation mask generated by the user and the history of the points traced by the user while drawing. It is composed of the fields listed in Snippet 1

The segmentation quality is a numeric value, between 0 and 1, that identifies the reliability of an action. In particular we consider the existence of malicious players, who might try to fool the rules of the game to achieve higher rewards and thus we need a measure

Snippet 1: Fields of a JSON Segmentation Object

```
{
        "quality": {
            "description": "Estimated quality of the segmentation",
            "type": "number"
        },
        "points": {
            "description": "Points used to reconstruct the binary mask for the segmentation",
            "type": "array",
            "items": {
                "x": "number",
                "y": "number"
            }
        },
        "history": {
            "description": "All the points that have been traced by the player",
            "type": "array",
            "items": {
                        "size": "number",
                        "color": "hex color format",
                        "points": {
                                        "type": "array",
                                        "items": {
                                                "x": "number",
                                                "y": "number"
                                        },
                                "time": "date"
            }
        }
}
```

to filter out misleading contributions. Segmentations are used to create binary masks that identify which pixels of the image belong to a given tag/cloth pair, as shown in Figure 5; the algorithm described in [6] is used to aggregate the gaming tracks and compute the best estimate of the mask, starting from all the available segmentations, using a filtered version of the segmentation data.

The history field contains the original, unfiltered sequence of points traced by the user; during the segmentation action, points are sampled with a constant period (e.g. 50ms) and sent to be stored together with the timing information representing the starting time. For this reason the history object is an array of objects containing points and time, stored together with size and colour of the current set of points. The number of points in each period varies depending on the user activity in that time interval. For each point the x and y coordinates are recorded. To correctly store the drawing action performed by a user, a timestamp is assigned to each point. In this way the drawing process can be easily replicated knowing the delay between consecutive points.

Table II reports details of the annotations collected from real human players over the course of 8 months related to a dataset of mixed images coming from established fashion datasets [18] [26].

Type of Data	Value
Images Analyzed	2396
Number of unique users	894
Avg Number of tags per Image	9.46
Segmentation Actions Submitted	15874
Number of matches	1381
Match Duration Avg.	423 seconds

TABLE II: Summary of the data collected through Sketchness

C. Implementation Strategy

The design of the bots requires some consideration on the activities that the artificial players should perform based on their role, in real time:

1)Sketcher: Emulating the human capabilities of a Sketcher is just a matter of replaying pre-recorded segmentation data submitted by the players respecting the timing information recorded from an earlier game session. The challenge of this approach resides in the fact that different segmentation actions could be available for the same image and only the best ones should be used by the bot; the problem of estimating the contributions' quality was solved as one of the steps necessary to aggregate the traces submitted by the players in a previous work [6] where a novel aggregation algorithm known as *Weighted Majority Voting* is used to associate a quality value to each available segmentation action and automatically estimate the reliability of human players.

To simulate a round using pre-recorded data, some basic information about the current image must be known: 1) The tag, which refers to the cloth object to be drawn. 2) At least one segmentation action performed by a user in a previous session. For this reason during a match which involves the use of bots, the tagging task, in which the sketcher is asked to provide a tag for the current image, is avoided and only annotated images will be used.

The necessary steps to emulate a Sketcher thus involve: (i) The retrieval of the segmentation actions associated to the current image. (ii) The extraction of the first n best segmentations, according to their quality, stored as a parameter of the segmentation and computed by the Weighted Majority Voting algorithm; in our scenario, n has been taken as the average number of segmentations per image, or if they were too few, all of the available ones. (iii) Random selection of one of the n best segmentations. (iv) The replay of the selected segmentation respecting the timings and other attributes such as trace width and colour.

The replay of the segmentation action ends either when the human player guesses the correct word, stopping the drawing routine, or when the whole trace has been reproduced, without any correct guess; in such case the Sketcher bot will be idle until the end of the round.

2)Guesser: To emulate a guesser it is not sufficient to rely on pre-recorded traces: even if the system knows which is the correct guess, human players usually try different guesses before answering correctly; if a human sketcher has drawn useless or wrong segmentations for the garment, the bot should not be able to recognize the hidden object.

The proposed solution exploits the fact that the images used in the game come from a known database, thus images' metadata can be extracted or computed a priori. The first key element to intelligently guess cloth names is the real-time retrieval of the current drawing position of the human sketcher, to find out the body part in which the traces are drawn and guess the possible clothes associated to that body area. For example, if the human player is drawing in the "feet" neighbour-

hood, possible guesses will be “shoes” or “socks”. The identification of the current drawing position is trivial, since all the positions are sent to the server through websockets packet updates.

By checking the location of the current drawing point of the Sketcher, it is possible to identify the current body part on which the Sketcher is drawing by relying on state of the art algorithms, such as tree-based models of part mixtures. [28]. A “Pose” metadata attribute for each image is used to store precomputed coordinates of the bounding boxes related to five different body parts “head” “torso” “arms” “legs” and “feet” , as shown in Figure 6. The images used for our experiments, coming from [26], already contained the validated metadata related to the pose, thus we skipped the problem of constructing reliable metadata associated to the pose; the framework is able to analyse and compute bounding boxes also for images never seen before.

Our intuition for identifying the specific garment drawn by the Sketcher is that the ratio between the area of the current segmentation and the area of the whole body of the subject could be used to discriminate among the possible garments in a given body part (for example in the “head” area the clothes could be “hats” or “glasses”).

The simulated guessing process, summarized in Figure 4 thus includes the following steps: (i) Retrieval of current drawing position. (ii) Classification of the current drawing position in a body part. (iii) Computation of the current area inside the segmentation traced by the user. (iv) Computation of the ratio between the current area drawn and the total area of the body in the image. (v) Given the ratio of the areas, guess the most appropriate cloth in the identified body part.

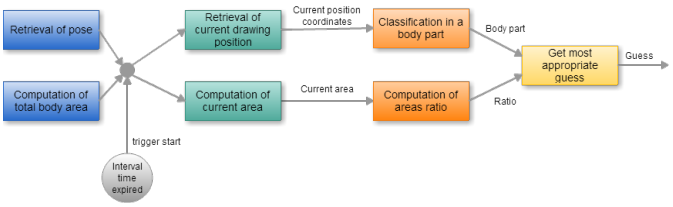

Fig. 4: Guesser Bot Process

The process is repeated every n seconds, until the guessing round terminates, either because the available time has expired or because the correct guess has been provided by the bot routines.

V. Experimental Analysis

To train the models used to classify a garment which contour is being traced by the human player in real time, seven supervised learning methods have been taken into consideration, namely (i) Naive Bayes [19], (ii) Bayes Networks [11], (iii) Neural Networks trained with backpropagation [17], (iv) Decision Trees [20], (v) Random Trees [21], (vi) Breiman’s Random Forests [7] and (vii) Support Vector Machines [23]. Naive Bayes compute probabilistic classifiers based on the assumption that all the variables (the data attributes) are independent. Bayes networks are probabilistic graphical models representing a set of random variables and their conditional dependencies via a directed acyclic graph. Neural Networks [17] are a widely used supervised learning method inspired by the early models of sensory processing by the brain. Decision trees are a well-known approach which produces human readable models represented as trees; Random Trees consider K randomly chosen attributes at each node, perform no pruning and have an option to allow estimation of class probabilities (or target mean in the regression case) based on a hold-out set (backfitting). Random Forests [7] are ensembles of decision trees that compute many decision trees from the same dataset, using randomly generated feature subsets and boostrapping, and generate a model by combining all the generated trees using voting. Finally, support vector machines are a rather recent method of machine learning based on structural risk minimization that proved to be very successful in solving complex classification and regression problems.

Fig. 5: Segmentation mask generated by the game

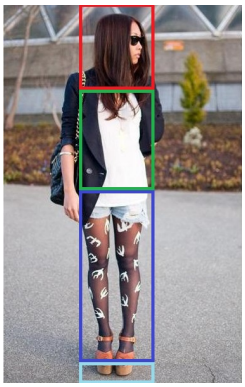

Fig. 6: Pose Estimation

A. Experimental Design

In our analysis we compared the performance of the aforementioned supervised learning techniques on the *Fashionista* dataset [26] composed by 686 manually annotated pictures collected from Chictopia.com, a social networking website for fashion bloggers; the annotations contained in the original dataset, including pose and cloths in images, has been used to create 5 different smaller dataset, one for each possible body part recognized by the body part detector. Each dataset is characterized by two attributes, a label identifying the cloth and a number identifying the ratio between the area of the garment’s binary mask and the whole body in the image; it is possible to identify automatically which dataset to choose based on the drawing position of the player with respect to the pose of the subject in the image, that is a precomputed value saved in the dataset.

The rationale behind our choice is that specific garments can be worn just on specific body parts (it is not useful to wear jeans on the arms) and that different

garments of the same kind occupy the same area with respect to the area of the body on which they are worn. A partial example of the "head" dataset in the *arff* format supported by Weka [14] is provided in Dataset 1. An initial exploratory analysis revealed that the classes distribution in the datasets was highly unbalanced, since some of the garments were rare and some of them could have been undistinguishable from each other since the players are able just to see the contour of the garment (e.g. jeans vs trousers). Thus, similar clothes were grouped together in the same class substituting the original tag; afterwards a resampling step (using the corresponding operator provided with Weka) was performed to rebalance the classes so as to generate new datasets with uniform distribution of garments. To compare the predictive performance of the seven supervised methods considered, we applied a 10-fold stratified crossvalidation using classification accuracy, both on the original datasets and the balanced ones.

Dataset 1 Weka Head Arff Dataset Excerpt

```
1: @relation clothsCorrelationInsideBodyPart
2: @attribute cloth glasses , necklace , scarf , hat ,
   earrings
3: @attribute ratio numeric
4: @data
5: glasses , 0.005872
6: necklace , 0.002764
7: scarf , 0.056912
```

B. Results and Discussion

We used the previously collected training set to build five decision models for each of the seven supervised learning methods; some of the models required additional preprocessing in order to improve the performance, by grouping together similar garments belonging to the same body part, in particular *torso* and *feet*. Then we compared the performances of the learned models by applying them (off-line) to the previously collected datasets. Both the training and the test of the decision models were carried out using the default parameters settings. The decision trees were learned by applying J48, a tree induction algorithm of Weka that works similarly to Quinlan's C4.5 [22]. To train the Naive Bayes classifier we used the Laplace correction [19] to prevent high influence of zero probabilities. In the case of random forests, the learned model consisted of ten random trees and the gain ratio was used as split criterion [15]. Concerning the neural networks, we used a feed-forward neural network with a single hidden layer trained by a backpropagation algorithm; the number of hidden nodes was set as $(|A| + |C|)/2 + 1$, where $|A|$ is the number of attributes and C is the number of classes; all the nodes use a sigmoidal activation function. Finally, for the Support Vector Machines we used the libSVM [8] implementation compatible with Weka with a *dot* kernel. Table IV compares the accuracy achieved by the models on the original datasets while Table V compares the accuracy achieved by the models on the balanced datasets with useless labels (e.g. hair, skin) removed; both the tables report also the number of instances for each considered body part.

As it is possible to see, the accuracy of the trained models on the original dataset depends on the particular body part on which they have been applied: some models, like the one related to the *feet* are able to achieve an accuracy of almost 80% on all the classification methods, while others, like the one related to the *torso* have accuracies ranging from 16% to 26% depending on the method. These results depend on several factors: first of all, our assumption that the ratio between the garment pixel area and the total body surface could be used to distinguish among different garments in the same body region was just an experimental hypothesis that needed to be validated. Moreover, in the same body region, some of the garments are very difficult to be recognized also by a human player if we consider just the shape of the contour. Finally, some of the garments present in the original dataset were extremely imbalanced w.r.t. the other ones. To solve the last two issues, the dataset was first balanced to keep the same ratio among the garments and similar garments were grouped together, as shown in Table III. Applying the classification methods over the revised dataset yielded the results shown in Table V; it is possible to note that one classification method outperforms all the others among almost all the possible body parts.

The low accuracy in identifying torso's garments is due to the fact that clothes which are apparently very different have similar dimensions. This is the case of dresses, bags and shirts/jackets. There is no way to solve this problem with the current solution, but this issue can be overcome with the guessing approach described later. The arms dataset is the one with less instances. Besides the lack of data, the error percentage is mainly due to the fact that the most of the items in this category are undistinguishable. However this is not a problem from the bot point of view: the task of distinguishing between a bracelet and a watch just by its contour is hard even for human players. The legs dataset is the one with the worst performances: the error percentage is due to the similarity between the few garments available in this category, such as shorts/skirt/pants. The feet category is instead the one with the best results. The classifier in this case has been reduced to a binary classifier, with an error percentage less than 10%.

C. Improving the Guessing Strategy

The previous approach has some drawbacks related to the guessing process: (i) If the human sketcher stops drawing, the area and the pointer position will not change anymore and the classifier will always return the same guess; since all the guess attempts are recorded to avoid duplicates then the game would enter in a stall condition. (ii) The merging process made during the dataset construction could prevent

the bot from guessing the correct cloth, since the solution could be represented by synonymous or tags similar to the one proposed by the classifier, that have been removed to improve the global performances. The guessing mechanics have thus been modified as illustrated in Figure 7.

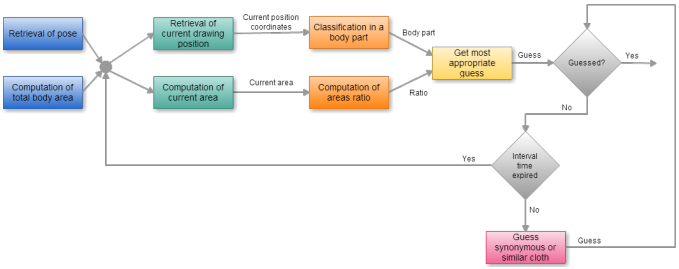

Fig. 7: Extended Guessing Strategy

The body part and ratio computations occur with a constant time period, to update its value the more the Sketcher traces the contours of the garment. During this time span, the classifier provides the most appropriate guess. If such guess is not the correct answer, the bot will attempt to guess synonymous labels or names of cloths similar to the one proposed by the classifier. This approach avoids game stalls, makes the bot appear more realistic and improves its probability of guessing the right word.

Dataset	Labels
head	glasses, (hair), necklace, scarf, hat, earrings
torso	blazer, t-shirt,blouse, bag, purse, sweater, shirt, jacket, top, cardigan, coat, dress, jumper, vest, accessories, cape, tie, bodysuit, wallet, romper, suit, bra, sweatshirt, intimate
torso (grouped)	shirt, bag, dress, tie, bodysuit, wallet, intimate
arms	(skin), purse, bracelet, watch, ring, gloves
legs	shorts, skirt, belt, pants, jeans
feet	tights, shoes, boots, leggings, socks, stockings, heels, sandals, wedges, flats, sneakers, loafers, clogs, pumps
feet (grouped)	socks, shoes

TABLE III: Labels of the classification datasets

VI. Evaluation

A fundamental aspect of each game that makes use of bots is how much human players consider them to be believable. To verify whether bots are indistinguishable from human players, Sketchness bots have been subjected to a “Turing-like” test. The test consisted of a single game with two human players and nine rounds. The subject plays a standard two players game, but the human opponent is randomly replaced by a bot player during some rounds of the match.

The predefined alternation between humans and bots was completely unknown to players, who were required to classify their opponent at the end of each round as human or artificial. Participants were of moderate skill range, with players neither ignorant to the game nor capable of playing as experts. The nine pairs of image-tag selected covered different scenarios of the game: first of all the choice of tags considered garments belonging to all the available body areas. A special attention was also put to the occurrence of incomplete-pose images, the ones in which the portrayed subject was not completely visible. When this event arises, the algorithm of Pose Estimation produces unexpected results, bringing to atypical and not human-like guesses coming from the bot player.

A statistical one-sample t-test has been performed, with the purpose of determining whether there was enough evidence to reject the hypothesis for which participants to the “Turing Test” answered randomly to the test questions. Statistics and critical values computed are shown in Table VI

Number of Samples	n	265
Samples Mean	$\overline{x}$	0.607
Standard Deviation	s	0.501
Degrees of Freedom (n-1)	d	264
Hypothetic mean	$\mu 0$	0.5
Significance level	α	5%
Computed t value	$t_{calc} = \frac{\overline{x}-\mu 0}{s/\sqrt{n}}$	3.491
Critical t value	$t_{\alpha/2,n-1}$	1.984

TABLE VI

Since $t_{calc} > t_{\alpha/2,n-1}$we can reject the hypothesis and state that participants to the test have not answered randomly to the provided questions, with a significance level of 5%. Let us now consider the results of the Turing Test, a two-class prediction problem, in which the outcomes are labelled either as “Bot” (positive) or “Human” (negative). Since the players submitted 39,26% of incorrect answers, in more than 33% of cases, the traditional threshold used in Turing-like tests, users were not able to distinguish correctly between human and bot players; we can thus state that the test was successful. Individual rates are listed in Table VII, where a True Positive (TP) is considered as a correctly identified bot.

The outcome shows that human players are easily identified by participants while classification errors are more prominent in the recognition of bots. This represent a meaningful result for our evaluation test since that proves the effectiveness of the approach that has been followed when designing the bots.

TP	89	TPR	0.593
FP	45	FPR	0.375
TN	75	TNR	0.625
FN	61	FNR	0.407

TABLE VII: Positive and negative predictive values and rates

VII. Conclusion

In this paper, we applied supervised learning techniques to design the artificial intelligence of a single player bot for a GWAP used to solve garment segmentation tasks in fashion images. After the analysis of the framework and the data available during the gameplay, the required steps for implementing the two possible game roles within a bot have been detailed. For the Sketcher emulation, replaying a pre-recorded,

Original Dataset	Head	Torso	Torso (Grouped)	Arms	Legs	Feet	Feet (Grouped)
Number of Instances	1018	1164	1164	1124	590	938	938
Multi Layer Perceptron (NN)	66.46	**26.92**	57.55	**71.52**	**59.49**	**40.19**	84.96
Naive Bayes	64.98	23.94	55.29	70.84	58.81	38.91	85.28
BayesNet	67.65	23.94	56.11	70.61	57.79	38.48	**85.82**
Decision Tree (J48)	**67.95**	23.22	**57.96**	70.16	57.28	38.91	85.71
LibSVM	35.31	16.75	50.05	56.94	43.89	32.40	78.89
RandomForest	55.49	18.91	49.23	59.68	52.71	27.61	78.35
RandomTree	54.89	17.98	48.71	58.08	52.37	26.97	78.25

TABLE IV: Classification Accuracy % over Original Dataset

Balanced Dataset	Head	Torso	Torso (Grouped)	Arms	Legs	Feet	Feet (Grouped)
Number of Instances	547	973	973	439	590	938	938
Multi Layer Perceptron (NN)	54.76	21.63	39.60	67.74	55.86	22.55	84.68
Naive Bayes	57.14	27.25	38.44	70.97	58.56	27.66	84.26
BayesNet	59.52	30.34	81.79	67.74	53.15	44.26	85.96
Decision Tree (J48)	67.86	57.58	81.21	64.52	53.15	65.11	85.53
LibSVM	24.03	11.92	24.87	18.68	41.18	10.13	80.06
RandomForest	78.57	67.13	87.28	70.97	67.57	**70.21**	89.36
RandomTree	**80.95**	**70.22**	**87.76**	**74.19**	**69.37**	69.79	**90.21**

TABLE V: Classification Accuracy % over Balanced and Filtered dataset

high quality segmentation action is sufficient to create the illusion of a human player tracing the contour of an object. Guessers required the introduction of a novel approach to emulate the perceptual and abstraction capabilities of human players. Starting from the Body Part segmentation of the subject portrayed in the image, it is possible to identify the region of the body in which a Sketcher is drawing. Once the body part has been identified, it is possible to choose the possible garment among a list of clothes typically worn in that region. Seven different supervised methods have been trained and tested over a dataset of images coming from fashion blogs; the goal was to recognize a specific garment based on the ratio between the area of the segmented cloth and the area of the overall body shape in the picture. Experimental results have shown that Random Trees are able to recognize garments based solely on the body part and the area of the submitted sketches with an accuracy that ranges from 70% to 90%. The trained models have also been evaluated in a Turing Test like scenario involving 30 players, showing that the proposed approach generates believable bots.

References

[1] Luis von Ahn. “Games with a Purpose”. In: *Computer* 39 (2006), pp. 92–94.

[2] Luis von Ahn et al. “Designing games with a purpose”. In: *Commun. ACM* 51 (2008), pp. 58–67.

[3] Luis von Ahn et al. “Improving Image Search with PHETCH”. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2007, Honolulu, Hawaii, USA, April 15-20, 2007*. 2007, pp. 1209–1212.

[4] Luis. von Ahn et al. “Labeling images with a computer game”. In: *SIGCHI*. Vienna, Austria, 2004.

[5] Luis von Ahn et al. “Verbosity: A Game for Collecting Common-sense Facts”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’06. Montreal, Quebec, Canada: ACM, 2006, pp. 75–78.

[6] C Bernaschina et al. “Robust aggregation of GWAP tracks for local image annotation”. In: *Proceedings of International Conference on Multimedia Retrieval*. ACM. 2014, p. 403.

[7] Leo Breiman. “Random Forests”. In: *Machine Learning* 45.1 (2001), pp. 5–32.

[8] Chih-Chung Chang et al. “LIBSVM: A library for support vector machines”. In: *ACM Transactions on Intelligent Systems and Technology* 2 (3 2011). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 27:1–27:27.

[9] Seth Cooper et al. “Predicting protein structures with a multiplayer online game”. In: *Nature* 466.7307 (2010), pp. 756–760.

[10] Seth Cooper et al. “The Challenge of Designing Scientific Discovery Games”. In: *Proceedings of the Fifth International Conference on the Foundations of Digital Games*. FDG ’10. Monterey, California: ACM, 2010, pp. 40–47.

[11] Professor Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks*. 1st. New York, NY, USA: Cambridge University Press, 2009.

[12] Luca Galli et al. “A Draw-and-Guess Game to Segment Images”. In: *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*. IEEE. 2012, pp. 914–917.

[13] Severin Hacker et al. “Matchin: eliciting user preferences with an online game”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’09. Boston, MA, USA, 2009.

[14] Mark Hall et al. “The WEKA Data Mining Software: An Update”. In: *SIGKDD Explor. Newsl.* 11.1 (Nov. 2009), pp. 10–18.

[15] Jiawei Han. *Data Mining: Concepts and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.

[16] Markus Krause et al. “Human computation games: A survey”. In: 2011.

[17] Miroslav Kubat. “Neural Networks: A Comprehensive Foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7.” In: *Knowl. Eng. Rev.* 13.4 (Feb. 1999), pp. 409–412.

[18] Babak Loni et al. “Fashion-focused Creative Commons Social Dataset”. In: *Proceedings of the 4th ACM Multimedia Systems Conference*. MMSys ’13. Oslo, Norway: ACM, 2013, pp. 72–77.

[19] Thomas M. Mitchell. *Machine Learning*. 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[20] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[21] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[22] Ross Quinlan. *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

[23] B. Schölkopf et al. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive computation and machine learning. MIT Press, 2002.

[24] Nitin Seemakurty et al. “Word sense disambiguation via human computation”. In: *Proceedings of the ACM SIGKDD Workshop on Human Computation*. Washington DC, 2010.

[25] Douglas Turnbull et al. “A game-based approach for collecting semantic annotations of music”. In: *In 8th International Conference on Music Information Retrieval (ISMIR*. 2007.

[26] K. Yamaguchi et al. “Paper Doll Parsing: Retrieving Similar Styles to Parse Clothing Items”. In: *Computer Vision (ICCV), 2013 IEEE International Conference on*. 2013, pp. 3519–3526.

[27] K. Yamaguchi et al. “Parsing clothing in fashion photographs”. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. June 2012, pp. 3570–3577.

[28] Yi Yang et al. “Articulated Pose Estimation with Flexible Mixtures-of-parts”. In: *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*. CVPR ’11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1385–1392.