Visual Objectification in Films: Towards a New AI Task for Video Interpretation

Supplementary Material

Table 3. Detailed list of films selected for creating dense annotations on objectification, with year of release and genre.

Film	Year	Genre(s)
Gone Girl	2014	drama, mystery, thriller,
Silver Linings Playbook	2012	drama, romantic, comedy
Crazy Stupid Love	2011	drama, romantic, comedy
The Help	2011	drama
Up in the Air	2009	drama, romantic, comedy
The Ugly Truth	2009	romantic, comedy
Marley and Me	2008	drama, family
Juno	2007	drama, comedy
Meet the Parents	2000	romantic, comedy
As Good As It Gets	1997	drama, romantic, comedy
Pulp Fiction	1994	drama, mystery
Sleepless in Seattle	1993	drama, romantic, comedy

We make available to the research community the contributed dataset *ObyGaze12*, as well as the code used to produce the results shown in this article and its supplemental material, at https://github.com/husky-helen/ObyGaze12.

7. Dataset

This section provides additional information on the list of films, the data annotation and processing procedure, and the calculation of γ Inter-Annotator Agreement (IAA).

7.1. List of films

The complete list of film of the *ObyGaze12* dataset is shown in Table 3. It corresponds to a 23%-subset of the MovieGraphs dataset [53]. The 12 movies we densely annotate for objectification construct and concepts were selected to approximately reproduce the fraction of genres in the original dataset.

7.2. Data annotation and processing

The data annotation and processing is illustrated in Fig. 5. During the annotation process, two annotators watch the film, and when they see a scene that is worth annotating, they freely indicate the boundaries of the scene, and then attribute an objectification level as well as concepts, resulting in the **Annotation 1** and **Annotation 2** timelines. Then during the data processing step, the annotations are projected onto the MovieGraphs delimitation (dashed gray lines), taking the highest level of objectification while enforcing a minimum overlap threshold of 20% (**Projection 1** and **Projection 2**). Annotations that have less than 20% overlap with the MovieGraphs delimitation are not taken into account (e.g., clips 1, 3, 4, and 5 of **Projection 1**), and when

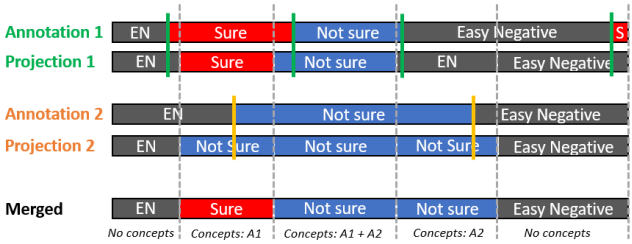

Figure 5. The annotation and data processing procedure is as follows. (1) Two experts annotate each film, with free delimitation (**Annotation 1** and **Annotation 2**). (2) Annotations are projected onto the MovieGraphs delimitation (dashed gray line), taking the highest level of objectification while enforcing a minimum overlap threshold of 20% (**Projection 1** and **Projection 2**). (3) Projections are **Merged**, taking the highest level of objectification and merging the concepts only for the same level of objectification.

multiple annotations have overlap $> 20\%$, the one with the highest level of objectification is kept (e.g., clips 2 and 4 of **Projection 2**). Finally, the projections are **Merged** to create a single timeline, taking the highest level of objectification and merging the concepts for the same level of objectification. The reason for this choice is that it appeared in the remediation session that most cases of initial disagreement were scenes that some annotators actually overlooked and agreed the objectification level should be raised to the maximum annotated, also considering concepts they had not noticed at first.

We generated multiple variations of the projections and merge by varying the minimum overlap threshold between 0.1-0.4. As the threshold increases, the numbers of projected and merged clips tagged with Sure and Not Sure decrease while those for Easy and Hard Negative increase, with an overall difference of $[+76, +68, -42, -103]$ clips for the four classes $[EN, HN, NS, S]$. An intermediate threshold of 0.2 was thus chosen for our experiments.

7.3. Inter-annotator agreement calculation

The γ Inter-Annotator Agreement [38] was designed to address the challenge of annotation tasks on a continuum without pre-defined units. It was motivated by text annotation tasks, but can be equally applied to similar tasks that involve both unitizing and categorization. The calculation reflects this by calculating the score of alignment and category separately:

$$d^{\alpha,\beta}_{combi(d_a,d_c)}(u,v) = \alpha.d_a(u,v) + \beta.d_c(u,v)\ , \quad (1)$$

where α and β represent the weights for the dissimilarities d_a and d_c in alignment and classification, respectively, for two annotations u and v in the annotation set $\mathcal{A}$. For our work, we calculate the γ value on the projected annotations, thus requiring only the d_c item (hence $d^{\alpha=0,\beta=1}_{combi(d_a,d_c)}(u,v)$) which is defined as a distance matrix between any two objectification levels, that we set to:

$$d_c(u,v) = \begin{Bmatrix} & EN & HN & NS & S \\ EN & 0 & 0.3 & 0.7 & 1 \\ HN & 0.3 & 0 & 0.4 & 0.7 \\ NS & 0.7 & 0.4 & 0 & 0.3 \\ S & 1 & 0.7 & 0.3 & 0 \end{Bmatrix} \quad (2)$$

The distances between all annotations in the movie are then averaged to obtain a disorder metric for the entire film:

$$\delta(a) = \frac{1}{\binom{4}{2}} \cdot \sum_{(u,v)\in\mathcal{A}^2} d(u,v) \quad (3)$$

In parallel, the dissimilarity is calculated and then averaged over N randomly generated sequences s to obtain a random disorder value for the corpus $\delta(c) = \frac{1}{N}\sum_{s\in c}\delta(s)$. γ is then calculated as:

$$\gamma = 1 - \frac{\delta(a)}{\delta(c)}\ , \quad (4)$$

where $\gamma \leq 0$ indicates random or worse. For our calculations, we used $N = 62$, at which γ has a confidence of $p < 0.01$.

Over every combination of pairs of annotators per film, we had in total 23 pairs of annotations which achieved an average of $\gamma = 0.42$, indicating a moderate level of agreement. Such a level is expected given the interpretive nature of the task and the low number of annotator per data sample. Recent works improve learning approaches by explicitly considering the IAA in cases of low number of annotators with moderate agreement [10, 55, 56]. Not considering the clips annotated Not Sure (NS), which is the uncertain and "noisy" class in human annotations, the IAA increases to $\gamma = 0.69$.

8. Experiments on task accuracy

8.1. Setup details

The implementation details of cross-validation and data balancing used to obtain the results presented in Table 2 are as follows. For each choice of negative set (EN or HN), each class is split into 10 equal-size folds. The last (resp. last but one) fold of each class is reserved for test (resp. validation), hence preserving class ratios. The remaining 8 folds of the positive class are used for train, while for each remaining 8 folds of negatives, a subset of folds is picked so as to obtain a balanced training set, the number of training sets depending on the class imbalance. The validation set allows to select the best model over training epochs for each training sets. The average performance of the models over the test set are shown in Table 2.

We keep the pre-trained models frozen and perform an adaptative max pooling of the resulting frame tokens, and feed the output to an MLP made of 2 dense layers, the hidden layer with 128 neurons and ReLU activations, the last with 2 softmax neurons. Experiments were carried out using a GTX 1080 Ti GPU, training of the MLP took approximately 1 hour and inference 30 minutes. Features extraction was performed with a GTX 1080 Ti GPU for 4 hours on average.

8.2. Random and all-positive baselines

In Table 2, we consider two trivial baselines independent of the data sample: *random* predicting positive with probability 0.5, and *random* predicting only positive. In such cases:

$$\text{precision} = F_{\text{data}} \quad ; \quad \text{recall} = F_{\text{classifier}}\ ,$$

where F_{data} is the fraction of positive samples in the test data, and $F_{\text{classifier}}$ is the fraction of samples predicted positive by the classifier. $F_{\text{classifier}}$ is 0.5 and 1 for *random* and *all-positive*, respectively. F_{data} is 23% and 19% for test sets EN vs. S and (EN U HN) vs. S, respectively. The resulting F1-scores are indicated for each trivial baseline and each test set in Table 2.

8.3. X-CLIP results on unseen movies versus unseen clips

In order to assess the feasibility of the task, the results presented in Table 2 are obtained when clips are split randomly between train, validation and test sets, as described in Sec. 8.1 above. Different clips from the same movie can therefore be in the training and test sets. It is hence possible that the X-CLIP adaptation presented in Table 2 result from overfitting on specific movies. We here test this hypothesis and consider distinct movies between train, test and validation sets.

We consider 10 movies for possible test and validation sets. Each test set is made of one of these movies. For each test set, validation sets are successively made of one of the 9 remaining movies. For each validation set, the training set is made of the remaining movies in the dataset. The training is made considering negative clip examples are HN only. Test is run on (ENUHN) vs. S clips. Other setup details are kept similar to those used to obtain the results shown in Table 2 and described in Sec. 8.1.

For each test set (movie), we present the average (and standard deviation) of the F1-scores obtained over all 9 best models for each validation set. Table 4 shows results averaged over all test sets and over every test movie. We ob-

serve that average F1-score is 0.53, to be compared with 0.82 in Table 2. The results show that the generalization over movies is harder than over clips only, and make for a future challenge to tackle.

Table 4. F1-score on each movie test, when movies in train, validation and test sets do not overlap.

Test movie	**F1-score**
As Good as it gets	0.55 (0.09)
Crazy, Stupid, Love	0.59 (0.05)
Gone Girl	0.54 (0.10)
Juno	0.67 (0.09)
Marley and Me	0.55 (0.07)
Pulp Fiction	0.29 (0.13)
Silver Linings Playbook	0.58 (0.03)
Sleepless in Seattle	0.51 (0.06)
The Help	0.56 (0.05)
Up in the Air	0.46 (0.05)
Average	0.53

9. Experiments on concept accuracy

Analysis of the decision tree Here we analyze the decision tree corresponding to the PCBM-DT model shown in Table 2 when the training set in Easy Negative vs. Sure. We remind that this decision tree is fed with the vector of similarities of the X-CLIP embedding of the clip to classify compared with every CAV. The CAV are obtained by training the SVMs on binary classification with negative examples being EN and positive examples being S and HN with the concept, as described in Sec. 4.2. The decision tree has a depth of 10 and the 4 first levels are shown in Fig. 6.

We first observe that a majority of child nodes on the left-hand side of their parent nodes correspond (i) to similarities with concepts lower than a threshold, and (ii) to a majority of negative samples. This is a consistent result, as the presence of a concept is conducive to a higher probability of an overall rating of objectification. Let us notice that this is not the case for the light-blue node with criterion *Expression of an emotion*, which shows this concept is likely not well captured by the X-CLIP embeddings. Second, with *Body* as root node, we observe that the presence of concept *Body* tends to structure the construct into two groups of occurrences of objectification: in the left-hand side sub-tree, when the concept tends to be absent, important discriminants are *Expression of an emotion*, *Look*, *Type of shot* and *Activities*; on the right-hand side sub-tree, when the *Body* concept tends to be present, important discriminants are *Posture*, *Clothing*, *Appearance* and *Activities*.

Beyond serving to analyze which concepts are currently poorly captured by existing models, the interpretable classifiers in a PCBM approach also serve film studies experts to analyze whether such groupings can corroborate existing theoretical analyses, or whether it is relevant to expand these analyses thanks to the newly identified groupings.

10. Experiments with an X-CLIP model pre-trained on LSMDC

In complement to the results in Table 2 of the X-CLIP model from [42] trained on the Kinetics dataset, we train another X-CLIP model introduced by Ma et al. [34] on the LSMDC movie dataset, following the procedure described in the code repository of [34]. Given the dissimilarity between Internet or instructional videos (such as those of Kinetics) and movies (noted, e.g., by [8]), our objective is to assess whether a model pre-trained on movie videos can achieve better performance at the new objectification-in-movie detection task. Following the guidelines in the code repository of [34], we retrained the X-CLIP model on the LSMDC dataset from scratch for 5 epochs. We used 4 RTX 8000 GPUs for 5 hours. Features extraction was performed with a GTX 1080 Ti GPU for 4 hours on average.

Table 5 presents the results of the model, obtained in the same condition as those presented in Table 2, to be compared with those of X-CLIP [42] pre-trained on Kinetics. We observe that the results are statistically equivalent, underlying the need for more efficient learning strategies to consider the specific concepts involved in the objectification occurrences.

Table 5. F1-score (average with standard deviations) obtained similarly as for Table 2 with the X-CLIP model of [42] re-trained on the LSMDC movie dataset.

Test	EN vs. S		(EN U HN) vs. S	
Train	EN vs. S	HN vs. S	EN vs. S	HN vs. S
X-CLIP [34] pre-trained on LSMDC	0.70 (0.08)	0.70 (0.10)	0.66 (0.06)	0.78 (0.11)

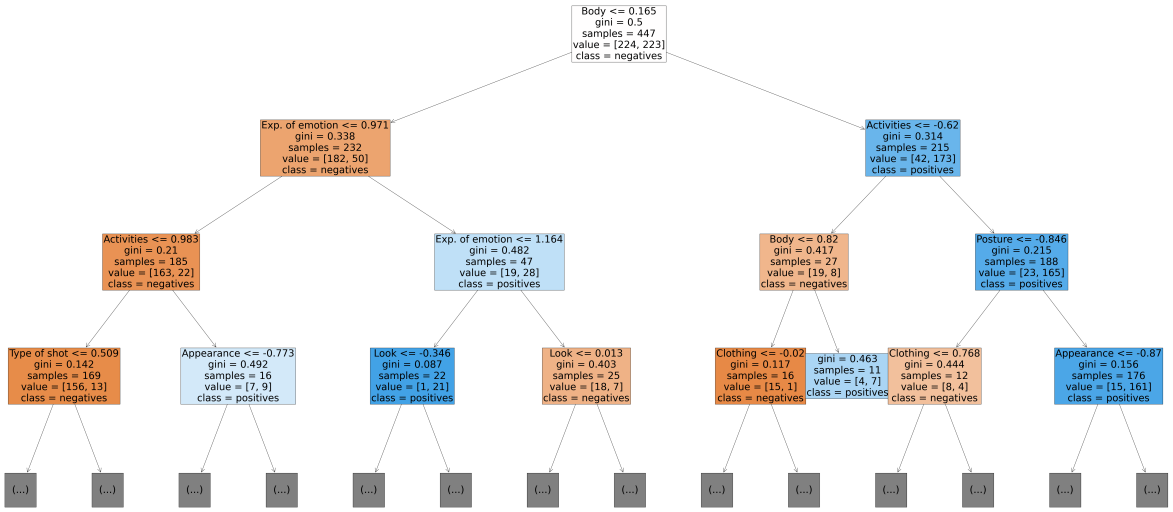

Figure 6. Decision tree trained for the objectification detection task of Easy Negative vs. Sure, fed with embedding similarities to CAV obtained from contrasting clips with concept against Easy Negative examples. Orange (resp. blue) shaded boxes represent a majority of negative (resp. positive) clip examples (i.e., without or with objectification).