LAEO-Net: revisiting people Looking At Each Other in videos

Manuel J. Marín-Jiménez
University of Cordoba
mjmarin@uco.es

Vicky Kalogeiton
University of Oxford
vicky@robots.ox.ac.uk

Pablo Medina-Suárez
University of Cordoba
i42mesup@uco.es

Andrew Zisserman
University of Oxford
az@robots.ox.ac.uk

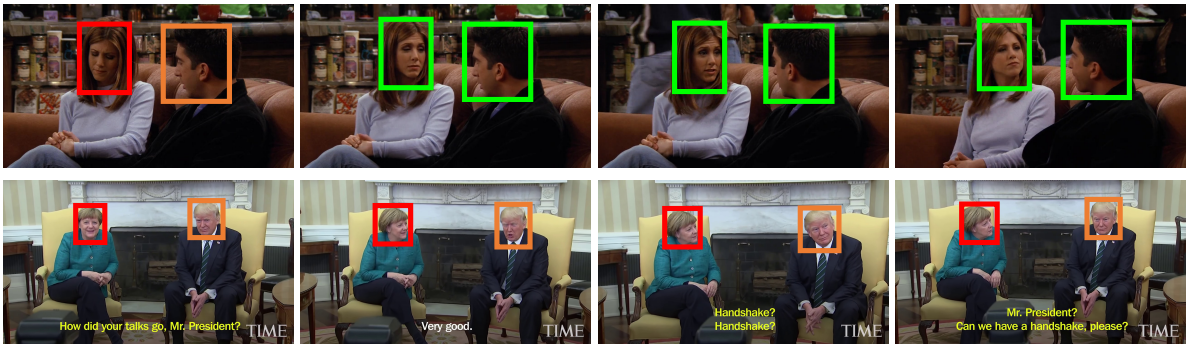


Figure 1: **Intimacy or hostility?** Head pose, along with body pose and facial expressions, is a rich source of information for interpreting human interactions. Being able to automatically understand the non-verbal cues provided by the relative head orientations of people in a scene enables a new level of human-centric video understanding. Green and red/orange heads represent LAEO and non-LAEO cases, respectively. Video source of second row: https://youtu.be/B3eFZMvNS1U

Abstract

Capturing the 'mutual gaze' of people is essential for understanding and interpreting the social interactions between them. To this end, this paper addresses the problem of detecting people Looking At Each Other (LAEO) in video sequences. For this purpose, we propose LAEO-Net, a new deep CNN for determining LAEO in videos. In contrast to previous works, LAEO-Net takes spatio-temporal tracks as input and reasons about the whole track. It consists of three branches, one for each character's tracked head and one for their relative position. Moreover, we introduce two new LAEO datasets: UCO-LAEO and AVA-LAEO. A thorough experimental evaluation demonstrates the ability of LAEO-Net to successfully determine if two people are LAEO and the temporal window where it happens. Our model achieves state-of-the-art results on the existing TVHID-LAEO video dataset, significantly outperforming previous approaches. Finally, we apply LAEO-Net to social network analysis, where we automatically infer the social relationship between pairs of people based on the frequency and duration that they LAEO.

1. Introduction

Eye contact or 'mutual gaze' is an important part of the non-verbal communication between two people [14]. The duration and frequency of eye contact depends on the nature of the relationship and reflects the power relationships, the attraction or the antagonism between the participants [1]. Therefore, in order to understand and interpret the social interactions that are occurring, it is important to capture this signal accurately. The importance of detecting people Looking at Each Other (LAEO) has already been recognized in a series of computer vision papers [16, 17, 20] as well as in other papers that study human gaze [3, 5, 22, 23].

LAEO is complementary to other forms of human non-verbal communication such as facial expressions, gestures, proxemics (distance), body language and pose, paralanguage (the tone of the voice, prosody), and interactions (e.g. hugging, handshake). Many of these have been the subject of recent papers [9, 15, 31].

In this paper, we introduce a new deep convolutional neural network (CNN) for determining LAEO in video material, coined **LAEO-Net**. Unlike previous works, our approach answers the question of whether two characters are LAEO over a temporal period by using a spatio-temporal model, whereas previous models have only considered individual frames. The problem with frame-wise LAEO is

that when characters blink or momentarily move their head, then they are considered non-LAEO, and this can severely affect the accuracy of the LAEO measurement over a time period. The model we introduce considers head tracks over multiple frames, and determines whether two characters are LAEO for a time period based on the pose of their heads and their relative position. Such an example is in Figure 1.

We make the following contributions: first, we introduce a spatio-temporal LAEO model that consists of three branches, one for each character's tracked head and one for their relative position, together with a fusion block. This is described in Section 3. To the best of our knowledge, this is the first work that uses tracks as input and reasons about people LAEO in the whole track, instead of using only individual frames. Second, we introduce two new datasets (Section 4): (i) **UCO-LAEO**, a new dataset for training and testing LAEO. It consists of 129 (3-12 sec) clips from four popular TV shows; and (ii) **AVA-LAEO**, a new dataset, which extends the existing large scale AVA dataset [9] with LAEO annotations for the training and validation sets. We evaluate the performance of the spatio-temporal LAEO model on both these new datasets (Section 6). Third, we show that our model achieves the state of the art on the existing TVHID-LAEO dataset [16] by a significant margin (3%). Finally, in Section 7, we show that by using LAEO scores we can compute a social network from character interactions in TV material, and we demonstrate this for two episodes of the TV comedy 'Friends'.

2. Related work

Gaze [22] and head pose [6] are powerful tools to deal with the problem of determining the *visual focus of attention* (VFoA) in a scene, *i.e*. what people are looking. For instance, [11] highlights the importance of the white part of the human eye (*i.e*. white sclera) in recognising gaze direction, enabling the extraordinary ability of humans to communicate between each other just by using gaze signals.

Visual focus of attention. One classical approach for determining the VFoA is [2], where the authors model the dynamics of a meeting group in a probabilistic way, inferring where the participants are looking at. More recently, the authors of [3] discover 3D locations of regions of interest in a video scene by analysing human gaze. They propose a probabilistic model that simultaneously infers people's location and gaze as well as the item they are looking at, which might even be outside the image. To evaluate their model, they record a total of 8 videos in controlled scenarios.

Gaze direction. Instead of videos, the work of [5] focuses on still images by proposing a neural network that estimates both the gaze direction and the VFoA, even if it is outside the image. A coarse spatial location of the target face in the image is provided to the network in the form of a one-hot vector. In contrast, in our model, the relative position of the human heads is provided by creating an RGB image with Gaussian-like circles representing the centre and scale of the heads and a colour-coding for indicating what is the target pair of heads to be analysed (Figure 5 (a). Therefore, our representation offers a better resolution of the scene geometry and incorporates information about head scales, providing a stronger input signal to the network.

Typically, in commercial movies, an action is represented among alternating video shots. Therefore, sometimes the VFoA is not visible in the current frame or shot, but in a different one. This problem is addressed in [23] with a deep learning model that reasons about human gaze and 3D geometrical relationships between different views of the same scene. The authors of [18] consider scenarios where multiple people are involved in a social interaction. Given that the eyes of a person are not always visible (*e.g*. due to camera viewpoint), they estimate people's gaze by modelling the motion of their heads with a Bayesian model, in which both gaze and VFoA are latent variables.

People Looking At Each Other. A special case of VFoA is when subject-A's VFoA is subject-B, and subject-B's VFoA is subject-A. This case is known as *mutual gaze* or people *looking at each other* (LAEO). This situation typically entails a non-physical human interaction; but might precede or continue a physical one, *e.g*. hand-shake before starting a conversation, or after finishing it. In the context of *Behaviour Imaging* research area, detecting LAEO events is a key component for understanding higher-level social interactions as, for example, in the study of autism in children [24].

The problem of detecting people LAEO in videos was first introduced in [16, 17]. After detecting and tracking human heads, [16] model and predict yaw and pitch angles of the human heads with a Gaussian Process regression model. Based on the estimated angles and the relative position of the two heads a LAEO score was computed per frame, and aggregated to provide a shot-level LAEO score. Even though we also model the head pose and relative position between two heads, our LAEO-Net differs from these works as it exploits the temporal consistency of neighbouring frames and estimates LAEO events for a track that spans over a temporal window, instead of a single frame.

In a controlled scenario with just two people interacting, [20] addresses the LAEO problem by using two calibrated cameras placed in front of the two participants, making sure that there is an overlapping visible zone between both cameras. It first estimates the eye gaze with a CNN and, then, uses 3D geometry to decide if the ray gaze of either participant intersects the head volume of the other one. In contrast, our goal is to address LAEO detection in general scenarios, with a (potentially) unrestricted number of subjects.

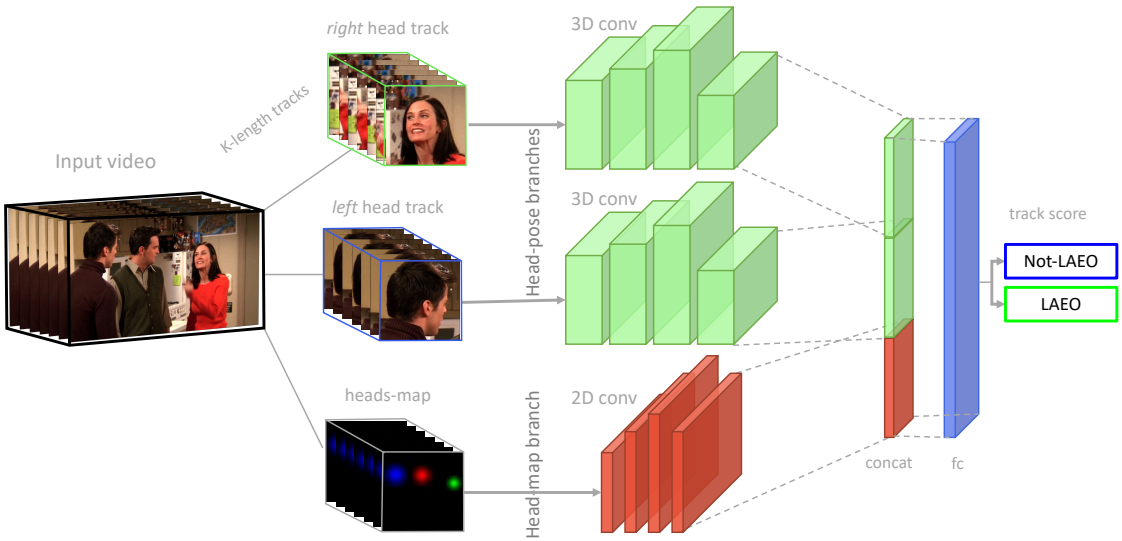


Figure 2: **Our three branch track LAEO-Net**: It consists of the head branches (green), the head-map branch (red) and a fusion block, which concatenates the embeddings from the other branches and scores the track sequence as LAEO or not-LAEO with a fully connected layer (blue) using softmax loss. In our experiments, we use head tracks of length $K = 10$.

branch details ('f': filter, 's': stride, $h \times w \times t$)	
head-pose	heads-map
input: frame crops 64×64×K	input: map 64×64
(four 3D conv layers)	(four 2D conv layers)
f: $16 : 5 \times 5 \times 3$, s:$2 \times 2 \times 1$	f: $8 \times 5 \times 5$, s:2×2
f: $24 : 3 \times 3 \times 3$, s:$2 \times 2 \times 1$	f: $16 \times 3 \times 3$, s:2×2
f: $32 : 3 \times 3 \times 3$, s:$2 \times 2 \times 1$	f: $24 \times 3 \times 3$, s:2×2
f: $12 : 6 \times 6 \times 1$, s:$1 \times 1 \times 1$	f: $16 \times 3 \times 3$, s:4×4

Table 1: **Architecture specification.**

3. LAEO-Net

Given a video clip, we aim to determine if any two humans are *Looking At Each Other* (LAEO). To this end, we introduce the LAEO-Net, a three branch *track* network, which takes as input two head tracks and the relative position between the two heads encoded by a head-map, and determines a confidence score on whether the two people are looking at each other or not, and the frames where LAEO occurs. The network is applied exhaustively over all pairs of simultaneous head tracks in the video clip.

LAEO-Net consists of three input branches, a fusion block, and a fully-connected layer and is illustrated in Figure 2. Two of the input streams determine the pose of the heads (green branches) and the third represents their relative position and scale (red branch). The fusion block combines the embeddings from the three branches and passes them through a fully-connected layer that predicts the LAEO classification (blue layer). The network uses spatio-temporal 3D convolutions and can be applied to the head tracks in the video clip. We next describe the components in detail and report their specifications in Table 1.

Head-pose branch. It consists of two branches, one per person. The input of each branch is a tensor of K RGB frame crops of size 64×64 pixels, containing a sequence of heads of the same person. Each branch encodes the head frame crop, taking into account the head pose. The architecture is inspired by the one proposed in [10] for shot boundary detection, with four 3D conv layers followed by a dropout and a flatten ones (green branches in Figure 2). The output of the flatten layer is L2-normalized before using it for further processing. Note that the head sequence of each person of the target pair will be encoded by this branch, obtaining two embedding vectors as a result.

Head-map branch. This branch embeds the relative position between two head tracks over time using a head-map. In particular, we define a 64×64 map to depict as 2D Gaussians all the heads detected in the central frame of the K-frames track (Figure 5 (a)). In addition to the two head tracks, this branch encodes information for other people in the scene. Depending on its size and scale, a third person could cut the *gaze ray* between the two side people (Figure 3). Including this information helps the LAEO-Net to distinguish such cases. This branch consists of a four 2D conv layers (Table 1). To obtain the embedding of the head-map we flatten the output of the last conv layer and apply L2-normalization.

Fusion block. The embedding vectors obtained as the output of the different branches of the network are concatenated and further processed by one fully-connected layer with an alternating dropout layer (blue layer in Figure 2). Then, a Softmax layer consisting of two hidden units (*i.e.* representing not-LAEO and LAEO classes) follows.

Overview of LAEO datasets						
statistics		UCO-LAEO (**new**)		AVA-LAEO (**new**)		TVHID-LAEO [16]
		train+val	test	train	val	test
#frames		$>$ 18k		$>$ 1, 4M (estim.)		$>$ 29k
#programs		4 (tv-shows)		298 (movies)		20 (tv-shows)
shots	#annotations	106+8	15	40166	10631	443
	#LAEO	77+8	15	18928	5678	331
pairs	#annotations	27358+5142	3858	137976	34354	–
	#LAEO	7554+1226	1558	19318	5882	–
sets (pairs)		32500	3858	137976	34354	443 (shots)

Table 2: **Summary of LAEO datasets.** *#programs*: different TV shows; *#shot-annotations*: annotated shots; *#shot-LAEOs*: shots containing at least one LAEO pair; *#pair-annotations*: annotated human bounding box pairs; *#pair-LAEOs*: human bounding box pairs that are LAEO; *sets:* #train/val/test LAEO pairs (or shots) used.

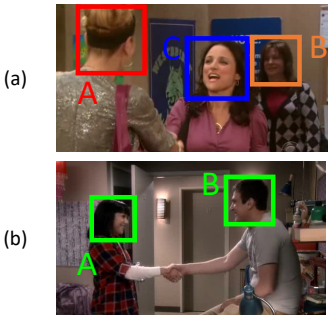


Figure 3: (a) AB are not LAEO as C is occluding. (b) AB are LAEO.

LAEO loss function. For training the LAEO predictor, we use the standard binary cross entropy loss:

$$\mathcal{L}_{\text{LAEO}} = -\left(c \cdot \log(\hat{p}_c) + (1-c) \cdot \log(1-\hat{p}_c)\right), \quad (1)$$

where c is the ground-truth class (*i.e.* 0 for not-LAEO, 1 for LAEO) and $\hat{p}_c$ is the predicted probability of the pair being class c.

4. Datasets

In this section, we describe the LAEO datasets. First, we introduce two new datasets: UCO-LAEO and AVA-LAEO, and then, two other datasets: AFLW [12], and TVHID [21]. AFLW is used for pre-training the head-pose branch and for generating synthetic data, while TVHID is used only for testing. The newly introduced UCO-LAEO and AVA-LAEO datasets are used both for training and testing the LAEO-Net. Table 2 shows an overview of the LAEO datasets.

The new datasets with their annotations and the code for evaluation are available online at: http://www.robots.ox.ac.uk/~vgg/research/laeonet/.

4.1. The UCO-LAEO dataset

We use four popular TV shows: 'Game of thrones', 'Mr Robot', 'Smallville' and 'The walking dead'. From these shows, we collect 129 (3-12 seconds long) shots and first annotate all the heads in each frame with bounding boxes, and then annotate each head pair as LAEO or not-LAEO. Some examples are shown in Figure 4 (top).

Annotation setup. We annotate all frames both at the frame level, *i.e.*, *does this frame contain any pair of people LAEO?*; and at the head level, *i.e.* we annotate all heads in a frame with a bounding-box and all the possible LAEO pairs. The visually ambiguous cases are assigned as 'ambiguous' and we exclude them from our experiments. Some examples are shown in Figure 4 (top). We split the 100 LAEO shots into 77 train, 8 validation and 15 test, respectively. This results in $\sim$7.5k training, $\sim$1.2k val and $\sim$1.5k test LAEO pairs (Table 2).

4.2. AVA-LAEO dataset

AVA-LAEO consists of movies coming from the training and validation sets of the 'Atomic Visual Actions' dataset (AVA v2.2) [9] dataset. The AVA frames are annotated (every one second) with bounding-boxes for 80 atomic actions, without containing LAEO annotations; therefore, we enhance the labels of the existing (person) bounding-boxes in a subset of the train and val sets with LAEO annotations.

Annotation setup. From the train and val sets of AVA, we select the frames with more than one person annotated as *'watch (a person)'*, resulting in a total of 40,166 and 10,631 frames, respectively. We only consider the cases, where both the watcher and the watched person are visible (since the watched person may not be visible in the frame).

For annotating, we follow the same process as in UCO-LAEO, *i.e.* we annotate each pair of human bounding boxes at the frame level as LAEO, not-LAEO, or ambiguous. This results in $\sim$19k LAEO and $\sim$118k not-LAEO pairs for the training set and $\sim$5.8k LAEO and $\sim$28k not-LAEO pairs for the val set (Table 2). We refer to this subset as AVA-LAEO. Figure 4 (bottom) shows some LAEO pair examples.

4.3. Additional datasets

AFLW dataset. We use the 'Annotated Facial Landmarks in the Wild' dataset [12] to (a) pre-train the head pose branch (first stage, Section 5.2), and (b) generate synthetic data for training (second stage, Section 5.3). It contains about 25k annotated faces in images obtained from FlickR, where each face is annotated with a set of facial landmarks. From those landmarks, the head pose (*i.e.* yaw, pitch and roll angles) is estimated. To create a sequence of head-crops, we replicate the input image K times. We keep the two middle replicas unchanged and randomly perturbing the others, *i.e.* small shift, zooming and brightness change.

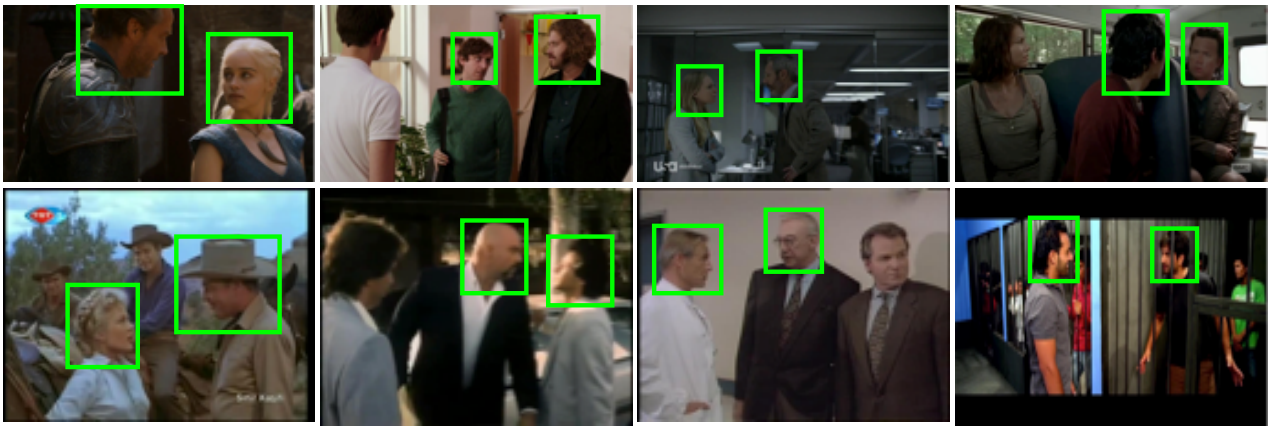


Figure 4: **(top) UCO-LAEO and (bottom) AVA-LAEO datasets.** Example of frames and LAEO head pair annotations included in our new datasets. Different scenarios, people clothing, background clutter and diverse video resolutions, among other factors, make them challenging.

TVHID-LAEO. The TVHID dataset [21] was originally designed for the task of human interaction recognition in videos. It contains 300 video clips with five classes: handshake, high-five, hug, kiss and negative. We use the LAEO annotations at the shot level from [16], which result in 443 shots with 331 LAEO and 112 not-LAEO pairs (Table 2).

5. Training LAEO-Net

LAEO-Net requires head tracks as input. Here, we first describe the head track generation (Section 5.1). Then, we describe our training procedure, which involves training in two stages. In the first stage (Section 5.2), we pre-train the head-pose branches and freeze their weights. In the second stage, we train the whole LAEO-Net (*i.e.* head-map branch and upper layers) from scratch (Section 5.3).

5.1. Head detection and tracking

Head detection. Our method requires head detections. In the literature, there are several models for face detection ([30] [32]); head detection, however, is a more challenging task, as it comprises detecting the whole head, including the face (if visible) but also the back of a head (*e.g.* [16]). We train a head detector using the Single Shot Multibox Detector (SSD) detector [13][1] from scratch. We train the model with a learning rate of 10^{-4} (first 50 epochs), and decrease it with a factor of 0.1 for the rest of the training. For speedup and better performance we use batch normalization and for robustness we use the data augmentation process from [13]. We train the head detector with the 'Hollywood heads' dataset [29]. It consists of head annotations for 1120 frames, split into 720 train and 200 val and test frames. We first train our detector with the training set and after validating the model, we train on the whole dataset as a refining stage.

Head tracking. Once we obtain head detections, we group them into tracks along time. For constructing tracks, we use the online linking algorithm of [27], as it is robust to missed detections and can generate tracks spanning different temporal extents of the video. Out of all head detections, we keep only the $N = 10$ highest scored ones for each frame. We extend a track T from frame f to frame $f + 1$ with the detection h_{f+1} that has the maximum score if it is not picked by another track and $\text{ov}\left(h_f^T, h_{f+1}\right) \geqslant \tau$, where ov is the overlap. If no such detection exists for M consecutive frames, the track stops; otherwise, we interpolate the head detections. The score of a track is defined as the average score of its detections. At a given frame, new tracks start from not-picked head detections. To avoid shifting effects in tracks, we track both forwards and backwards in time.

5.2. Pre-training the head-pose branches

In general, humans can infer *where* a person is looking just based on the head pose, without even seeing the eyes of the person. This shows that most information is encoded in the head orientation. There are several works that model the head orientation [26] or the eye gaze [22]. Note that using the actual eye gazing is not always an option, even with multiple-frames as input, as there is no guarantee that the eyes are fully visible, *i.e.* due to image resolution, or self occlusions. Therefore, in this work, we model gaze just based on head orientation.

We model head orientation with three angles (in order of decreasing information): (a) the yaw angle, *i.e.* looking right, left, (b) the pitch angle, *i.e.* looking up, down, and (c) the roll angle, *i.e.* in-plane rotation. We use this modelling to pre-train the head-pose branches. We use the

[1] Detector: https://github.com/AVAuco/ssd_people

weights learnt from this pre-training without tuning this branch further (freezing the weights).

Loss function of head-pose pre-training. Let (α, β, γ) be the yaw, pitch and roll angles of a head, respectively. We define one loss for estimating each pose angle: $\mathcal{L}_\alpha$, $\mathcal{L}_\beta$, $\mathcal{L}_\gamma$ and model them with the $L1$-smooth loss [25].

Given that the yaw angle is the dominant one, in addition to these losses, we include a term that penalizes an incorrect estimation of the sign of the yaw angle, *i.e.*, failing to decide if the person is looking left or right ($\mathcal{L}_s$). It is defined as:

$$\mathcal{L}_s = \max(0, -\text{sign}(\alpha) \cdot \text{sign}(\hat{\alpha})), \tag{2}$$

where $\text{sign}(\alpha)$ is the sign function (*i.e.* returns $+1$ for positive inputs, -1 for negative inputs, and 0 if the input is 0) applied to the yaw angle; and, $\hat{\alpha}$ is the ground-truth angle.

Therefore, the loss function $\mathcal{L}_h$ for training the head-pose branch for LAEO purposes is given by:

$$\mathcal{L}_h = w_\alpha \cdot \mathcal{L}_\alpha + w_\beta \cdot \mathcal{L}_\beta + w_\gamma \cdot \mathcal{L}_\gamma + w_s \cdot \mathcal{L}_s, \tag{3}$$

where w_x are positive weights chosen through cross-validation during training. In our experiments, we use: $w_\alpha = 0.6$, $w_\beta = 0.3$, $w_\gamma = 0.1$, $w_s = 0.1$, as w_α is the dominant one determining the head orientation. Note that the weights do not necessarily add to 1.

5.3. Training the LAEO-Net

We train the LAEO-Net with both real and synthetic data. We also use **data augmentation** techniques, such as image perturbations, translations, brightness changes, zoom changes, *etc*. For the first $N = 2$ epochs, we train the LAEO-Net only with synthetic data, and then at each **training step** we alternate between real and synthetic data. To improve the performance of the model, we also use hard negative mining. We deploy the **curriculum learning** strategy of [19], which facilitates the learning by modulating the difficulty of the hard negatives incorporated into the training phase. In our experiments, the value of the hyper-parameter τ is increased after 2 epochs, allowing more difficult examples as its value increases. The importance of our training scheme is evaluated in Section 6.2.

Synthetic data. For generating additional synthetic data we use images with head-pose information. To generate positive samples, we select pairs of head images whose angles are compatible with LAEO and, at the same time, they generate consistent geometrical information. To generate negative samples, we (i) change the geometry of the pair, *i.e.* making LAEO not possible any more, for instance, by mirroring just one of the heads of the pair or (ii) select pairs of head images whose pose are incompatible with LAEO, *e.g.* both looking in the same direction. Figure 5 (b) shows some artificially generated examples.

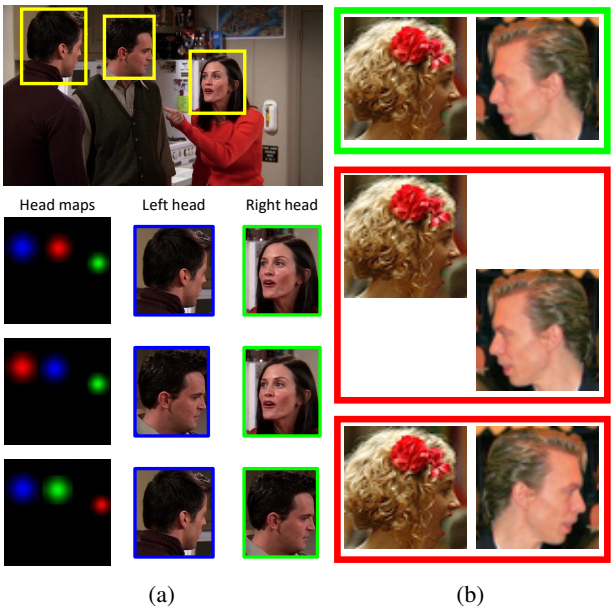


Figure 5: **(a) Head-maps and (b) augmentation of LAEO training samples.** (a) For each frame, we analyse all head pairs, using a color coding: *blue* for the left, *green* for the right and *red* for the remaining heads, such as middle (*i.e.* not currently considered for evaluation). (b) We generate additional synthetic LAEO negative training data (red boxes) from each positive pair (green box), based on the orientation or the relative position of the heads.

6. Experimental results

In this section, we experimentally evaluate the effectiveness of LAEO-Net for determining people LAEO. Note that the model is trained either on the UCO-LAEO or on the AVA-LAEO dataset. We, first, report the evaluation protocol (Section 6.1), and we assess the importance of synthetic data during fine-tuning (Section 6.2). Then, we examine the impact of each component of the model (Section 6.3), and in Section 6.4 we examine the performance of LAEO-Net on the two new test datasets, UCO-LAEO and AVA-LEO. Finally, we compare our LAEO-Net to the state-of-the-art methods on the TVHID-LAEO dataset (Section 6.5).

Implementation details. LAEO-Net is implemented with Keras [4] using TensorFlow as backend. For training the LAEO-Net we use the Adam optimizer with mini-batches of 9 samples: 4 positive, 4 negative and 1 hard negative. The learning rate starts at 10^{-4} and is reduced by a factor of 0.2 when the AP on the validation set does not improve for 5 consecutive iterations. The minimum possible learning rate is set to 10^{-8}. Dropout is set to 0.5. Before applying the trained model to the test sets, the samples of the validation set of UCO-LAEO are added to the training set for few additional epochs using the latest learning rate.

6.1. Evaluation protocols and scoring methodology

LAEO-classification AP is the metric we use to evaluate the LAEO predictions. Similar to object detection, a detection is correct if its intersection-over-union (IoU) overlap with the ground-truth box is > 0.5 [7]. A detected pair is correct if both heads are correctly localized and its label (LAEO, not-LAEO) is correct. The performance is Average Precision (AP) computed as the area under the Precision-Recall (PR) curve. Depending on the available ground-truth annotations, we measure AP at frame level, considering each pair as an independent sample, or at shot-level, if more detailed annotations are not available. Frame level is used for *UCO-LAEO* and *AVA-LAEO* and, following previous work [16, 18], shot level for *TVHID*.

Scoring methodology. Given that the level of (ground truth) annotation differs between the three datasets, we describe how we use the LAEO-Net outputs to obtain the final scores (either at the shot or at the frame level). To this end, we test the LAEO-Net on pairs of head-tracks (of length $K = 10$) and obtain one LAEO score for each track-pair. We assign the LAEO score to the head-pair in the middle frame. We describe below the scoring process for each dataset.

i. *UCO-LAEO:* Since the bounding boxes for the heads are available for each frame, the LAEO-Net is applied directly to these head tracks (no detections are used). To account for the $K/2$ frames at the beginning (resp. end) of a track, we propagate the score from the middle frame.

ii. *AVA-LAEO:* We run the head tracker and apply the LAEO-Net on these tracks. Given that AVA-LAEO contains pair annotations for *human* bounding-boxes (instead of heads), we compare each detected head pair against the ground-truth human pairs using intersection over head area (instead of IoU).

iii. *TVHID:* We run the head tracker and apply the LAEO-Net on these tracks. We compute a LAEO score as the maximum of the smoothed scores in a shot; the smoothed score of each pair is the average of a moving temporal window (of length five) along the track.

6.2. Importance of synthetic data

Using synthetic data (*i.e.* change of relative position of heads to create LAEO and not-LAEO pairs, see Figure 5 (b) allows for more training samples for free, thus making the model more generalizable while reducing the probability of overfitting (Section 5.3). Training and testing LAEO-Net on UCO-LAEO results in an AP=79.5%, while training: (i) without synthetic data and without hard negative mining results in AP=64.8%, *i.e.* 14.7% decrease, (ii) with synthetic data but without hard negative mining results in AP=70.2%, *i.e.* 9.3% decrease, (iii) without synthetic data but with hard negative mining results in AP=71.2%, *i.e.* 8.2% decrease, (iv) with only synthetic data (no real data and no hard negative mining) results in AP=76.9%, *i.e.* 2.6% decrease. These results reveal the significance of using synthetic data in the training process.

head pose +	LAEO %AP on UCO-LAEO		
head-map	K=1 (2D)	K=5	**K=10**
-	64.9	58.7	73.5
✓	72.7	73.9	**79.5**

Table 3: **Ablation study of LAEO-Net**. We report %AP of LAEO-Net when trained and tested on UCO-LAEO for various temporal windows K=1,5,10.

6.3. Ablation study

LAEO-Net consists of the head-pose and the head-maps branches. Here, we study the impact of some architecture choices, in particular, the head-maps branch and the length of the temporal window K (Table 3).

We evaluate the LAEO-Net without and with the head-map branch. We observe that adding the heap-map branch improves the performance of our architecture (from 73.5% to 79.5% for $K = 10$), as it enables learning the spatial relationship between the two heads. Moreover, to assess the importance of the temporal window using K frames compared to using a single frame, we vary K and train and evaluate the LAEO-Net with $K = 1, 5, 10$. Table. 3 shows that there is an improvement in AP performance (of 1.2%) when K increases from only 1 to 5 frames, and a significant improvement (of 6.8%) when K increases from only 1 to 10 frames. In this work, we use $K = 10$ frames. Training the LAEO-Net without freezing the weights of the head-pose branch results in AP=75.1% vs AP=79.5%, demonstrating that freezing weights results in performance improvements.

Comparison with an alternative architecture. The core branches of LAEO-Net are the head-pose branches. In addition, we use the head-map branch to describe the relative position between two heads in the tracks over time. Here, we consider one alternative to the head-map, the *geometrical information branch*, where the relative position of two heads over time is encoded based on their geometry.

The *geometrical information branch* embeds the relative position between two head tracks over time (relative to a $(1, 1)$ normalized reference system), as well as the relative scale of the head tracks. The input is a tuple (dx, dy, s_r), where dx and dy are the x and y components of the vector that goes from the left head L to the right one R, and $s_r = s_L/s_R$, is the ratio between the scale of the left and right heads. The encoding network consists of two fully-connected layers with 64 and 16 hidden units, respectively. Therefore, it outputs a vector of 16 dimensions encoding the the geometrical relation between the two target heads.

When we replace the head-map branch with the geometry branch, for UCO-LAEO and $K = 10$, the classification AP of the LAEO-Net with the *geometry branch* is is between $3 - 5\%$ less than the AP with the head-map branch.

LAEO % AP							
train on	UCO	AVA	UCO	AVA	UCO	AVA	TVHID
test on	UCO-LAEO		AVA-LAEO		TVHID		
baseline (chance level)	40.4		17.1		–		
[16] (Fully auto+HB)	–	–	–	–	–	–	87.6
[18] (Fine head orientation)	–	–	–	–	–	–	89.0
LAEO-Net	**79.5**	77.8	39.1	**50.6**	**91.8**	90.7	–

Table 4: **LAEO results on UCO-LAEO, AVA-LAEO and TVHID.** We report %AP at the pair@frame level for TV-LAEO and AVA-LAEO and, similar to other works, at the shot level for TVHID.

This is expected; even though both branches encode the same information (*i.e*. relative position of the two heads), the head-maps branch provides a richer representation of the scene and, therefore, results in better AP. Note that using both the head-map *and* the geometry branches (in addition to the head-pose branches) does not lead to any further performance improvement, due to the fact that the combination of these two branches just increases the number of network parameters without providing any additional information. Therefore, we conclude that the proposed LAEO-Net is the most effective architecture in terms of AP performance.

6.4. Results on UCO-LAEO and AVA-LAEO

We evaluate the LAEO-Net on the UCO-LAEO and AVA-LAEO datasets and report the results in Table 4. When training and testing on UCO-LAEO, the performance is 79.5%, demonstrating the effectiveness of our model. When training and testing on AVA-LAEO, the performance is 50.6%. These results reveal that there exists a significant gap in the performance between UCO-LAEO and AVA-LAEO. This is due to the different nature of AVA-LAEO compared to other datasets: (1) head annotations are not provided (just human bounding-boxes every 1 second); (2) it contains challenging visual concepts, such as (a) low resolution movies, (b) many people in a scene, (c) blurry, small heads, and (d) particular clothing styles, *e.g*. several people wearing hats (western, Egyptian's, turbans, *etc*.). Despite these difficulties, the LAEO-Net achieves AP=50.6% on AVA-LAEO.

To examine the generalization of LAEO-Net to other datasets, we also report results when training and testing with different datasets, *i.e*., AP=77.8% for UCO-LAEO and AP=39.1% for AVA-LAEO. These results show that the domain shift [28] definitely affects the performance, but at the same time our models are able to generalize to other unseen datasets. To assess the difficulty of these datasets and the effectiveness of LAEO-Net, we also compare it to the chance level classification, LAEO-Net outperforms chance level by a large margin: +40% for UCO-LAEO and +33% for AVA-LAEO.

When applying LAEO-Net on UCO-LAEO and AVA-LAEO we obtain the results of Figure 6. For both datasets, we display some of the highest ranked pairs of people that are LAEO. We observe that the LAEO-Net leverages the head orientations and their temporal consistency and accurately determines the frames where people are LAEO.

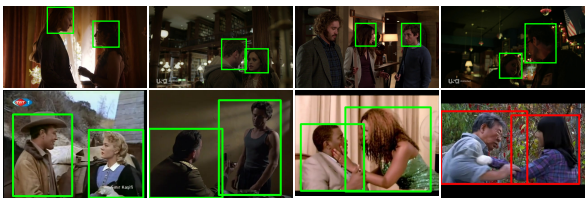


Figure 6: **LAEO-Net results on (top) UCO-LAEO and (bottom) AVA-LAEO.** For different scenarios, backgrounds, head poses *etc*., in most cases the LAEO-Net successfully determines if two people are LAEO (green boxes); in some other cases (red boxes), only by tracking the eye movement, we could determine if the people are LAEO.

When applying LAEO-Net on UCO-LAEO and AVA-LAEO we obtain the results of Figure 6. For both datasets, we display some of the highest ranked pairs of people that are LAEO. We observe that the LAEO-Net leverages the head orientations and their temporal consistency and accurately determines the frames where people are LAEO.

We hope that the LAEO-Net with these two datasets will provide solid baselines and help future research on this area.

6.5. Results on TVHID-LAEO

We compare LAEO-Net to the state of the art on TVHID [21], *i.e*. the only video dataset with LAEO annotations (Section 4.3). As in [16], we use average AP over the two test sets of TVHID (Table 4). On this dataset, LAEO-Net trained on UCO-LAEO and AVA-LAEO achieves AP= 91.8% and AP=90.7%, respectively. Both these results outperform all other methods by a large margin ($2-3\%$).

We apply the LAEO-Net on TVHID and obtain the results shown in Figure 7. Our model successfully detects people LAEO in several situations and scenarios, such as different illuminations, scales, cluttered background. By examining the remaining 8% error, we note that in most cases, the ground truth label is ambiguous (first two red frames in Figure 7). In some cases though, the head pose and relative position are not sufficient cues to model LAEO cases, because the LAEO event can be determined only by examining the eye gaze (last red frame in Figure 7). Our method struggles with such difficult cases, which are typically handled by eye-tracking techniques. A possible extension could be to include an eye-tracking branch.

7. Social network analysis: *Friends*-ness

One principal way of signaling an interest in social interaction is the willingness of people to LAEO [8, 14]. The duration and frequency of eye contact reflects the power relationships, the attraction or the antagonism between people [1]. Here, we present an application of our LAEO-Net to a social scenario: given head tracks we automatically infer relationships between people (*e.g*. liking each other, ro-



Figure 7: **LAEO-Net results on TVHID.** The top three rows depict correct LAEO results when the ground truth is LAEO (green) and not-LAEO (blue). LAEO-Net successfully detects people LAEO in several situations (different illuminations, scales, cluttered background). Most failure cases are missing people LAEO in ambiguous scenes (first two red frames). In the last red frame though we see a difficult failure case, where LAEO-Net predicts a not-LAEO sequence as LAEO. Note that the head orientation and posture points to LAEO; the character, however, rolls his eyes!

mantic relationships) based on the frequency and duration that people LAEO over time. In particular, the idea behind *Friends-ness* is to compute the ratio between the number of frames that people are LAEO over the frames that they share a scene. The higher the ratio, the more they interact.

Dataset and procedure. For this experiment, we use two episodes of the TV show 'Friends' (*s03ep12* and *s05ep14*). First, we detect and track all heads, which results in almost $3k$ head tracks. Then, without any further training, we apply the LAEO-Net on each track pair to determine if two characters are LAEO.

Character annotation. To determine which head track corresponds to which character, we annotate all produced head tracks as depicting one of the main characters (more than half of the tracks), irrelevant characters ($\sim$30%), being wrong (15%) or some secondary ones (the rest).

Experiments and discussion. Our goal is to automatically understand underlying relationships between characters. Therefore, we measure the 'likeness' between two characters as the average LAEO score over the frames two characters co-exist, and depict it in Figure 8 (the thicker the edge, the higher the score and the stronger the relationship). We observe that the LAEO score captures the dominant relationships between characters, *e.g.* Ross and Rachel vs characters that are more distant, *e.g.* Phoebe and Chandler. Our study reveals all prominent pair relationships, demonstrating that the more people are LAEO, the stronger their *interaction* and *social relationship*.

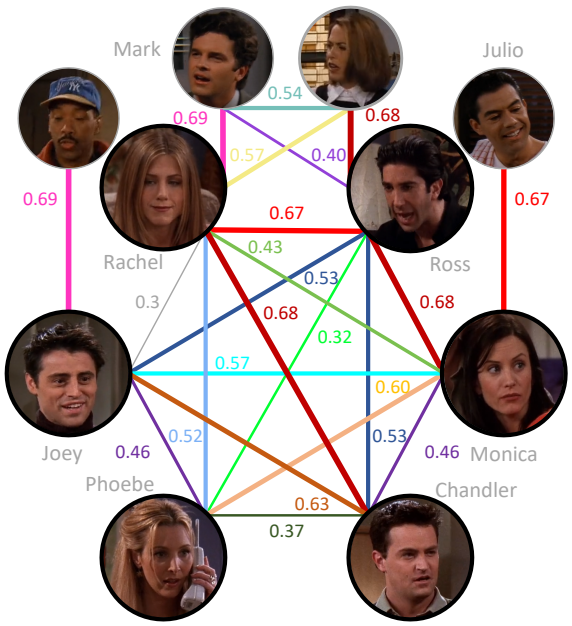


Figure 8: ***Friends*-ness on Friends.** We measure 'friends-ness' with the average LAEO score of each character pair and depict it with the edges in the graph: the thicker the edge, the more dominant the relationship. Some patterns are clear: Ross and Rachel or Monica and Julio 'like' each other more than Chandler and Phoebe or Ross and Mark. (*Best viewed in digital format.*)

8. Conclusions

In this paper, we focused on the problem of people *looking at each other (LAEO)* in videos. We proposed LAEO-Net, a deep *track* architecture, which takes as input head tracks and determines if the people in the track are LAEO. This is the first work that uses *tracks* instead of bounding-boxes as input to reason about people on the whole track. LAEO-Net consists of three branches, one for each character's tracked head and one for the relative position of the two heads. Moreover, we introduced two new LAEO video datasets: UCO-LAEO and AVA-LAEO. Our experimental results showed the ability of LAEO-Net to correctly detect LAEO events and the temporal window where they happen. Our model achieves state-of-the-art results on the TVHID-LAEO dataset. Finally, we demonstrated the generality of our model by applying it to a social case scenario, where we automatically infer the social relationship between two people based on the frequency they LAEO.

Acknowledgements. We are grateful to our annotators: RF, RD, DK, DC and EP, and to NVIDIA for donating some of the GPUs used in this work. This work was supported by the Spanish grant "José Castillejo", the EPSRC Programme Grant Seebibyte EP/M013774/1, and the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number D17PC00341.

References

[1] A. Abele. Functions of gaze in social interaction: Communication and monitoring. *Journal of Nonverbal Behavior*, 1986. 1, 8

[2] S. O. Ba and J.-M. Odobez. Recognizing visual focus of attention from head pose in natural meetings. *IEEE Trans. on Systems, Man, and Cybern., Part B*, 2009. 2

[3] E. Brau, J. Guan, T. Jeffries, and K. Barnard. Multiple-gaze geometry: Inferring novel 3D locations from gazes observed in monocular video. In *ECCV*, 2018. 1, 2

[4] F. Chollet et al. Keras. https://keras.io, 2015. 6

[5] E. Chong, N. Ruiz, Y. Wang, Y. Zhang, A. Rozga, and J. M. Rehg. Connecting gaze, scene, and attention: Generalized attention estimation via joint modeling of gaze and scene saliency. In *ECCV*, 2018. 1, 2

[6] V. Drouard, R. Horaud, A. Deleforge, S. Ba, and G. Evangelidis. Robust head-pose estimation based on partially-latent mixture of linear regressions. *IEEE Transactions on Image Processing*, 2017. 2

[7] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 Results, 2007. 7

[8] E. Goffman. *Behavior in public places*. Simon and Schuster, 2008. 8

[9] C. Gu, C. Sun, D. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 1, 2, 4

[10] M. Gygli. Ridiculously fast shot boundary detection with fully convolutional neural networks. *CoRR*, abs/1705.08214, 2017. 3

[11] H. Kobayashi and S. Kohshima. Unique morphology of the human eye and its adaptive meaning: comparative studies on external morphology of the primate eye. *Journal of human evolution*, 2001. 2

[12] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization. In *Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011. 4

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 5

[14] B. K. Loeb. Mutual eye contact and social interaction and their relationship to affiliation. 1972. 1, 8

[15] M. J. Marín-Jiménez, R. Muñoz Salinas, E. Yeguas-Bolivar, and N. Pérez de la Blanca. Human interaction categorization by using audio-visual cues. *Machine Vision and Applications*, 2014. 1

[16] M. J. Marín-Jiménez, A. Zisserman, M. Eichner, and V. Ferrari. Detecting people looking at each other in videos. *IJCV*, 2014. 1, 2, 4, 5, 7, 8

[17] M. J. Marín-Jiménez, A. Zisserman, and V. Ferrari. "Here´s looking at you, kid". Detecting people looking at each other in videos. In *BMVC*, 2011. 1, 2

[18] B. Massé, S. Ba, and R. Horaud. Tracking Gaze and Visual Focus of Attention of People Involved in Social Interaction. *IEEE PAMI*, 2018. 2, 7, 8

[19] A. Nagrani, S. Albanie, and A. Zisserman. Learnable PINs: Cross-modal embeddings for person identity. In *ECCV*, 2018. 6

[20] C. Palmero, E. A. van Dam, S. Escalera, M. Kelia, G. F. Lichtert, L. P. Noldus, A. J. Spink, and A. van Wieringen. Automatic mutual gaze detection in face-to-face dyadic interaction videos. In *Proceedings of Measuring Behavior*, 2018. 1, 2

[21] A. Patron-Perez, M. Marszałek, A. Zisserman, and I. D. Reid. High five: Recognising human interactions in TV shows. In *BMVC*, 2010. 4, 5, 8

[22] A. Recasens, A. Khosla, C. Vondrick, and A. Torralba. Where are they looking? In *NIPS*, 2015. 1, 2, 5

[23] A. Recasens, C. Vondrick, A. Khosla, and A. Torralba. Following gaze in video. In *ICCV*, 2017. 1, 2

[24] J. M. Rehg. Behavior imaging: Using computer vision to study autism. *Machine Vision and Applications*, 2011. 2

[25] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 6

[26] N. Ruiz, E. Chong, and J. M. Rehg. Fine-grained head pose estimation without keypoints. In *CVPR Workshops*, 2018. 5

[27] G. Singh, S. Saha, M. Sapienza, P. H. Torr, and F. Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *ICCV*, 2017. 5

[28] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 8

[29] VGG. Hollywood heads dataset. http://www.robots.ox.ac.uk/~vgg/software/headmview/, 2011. 5

[30] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 5

[31] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating visual representations from unlabeled video. In *CVPR*, 2016. 1

[32] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012. 5