Seeing the Arrow of Time

Mathilde Bateson
ENS Paris-Saclay
mathilde.bateson@gmail.com

Jules Scholler
ENS Paris-Saclay
jules.scholler@gmail.com

Abstract

We explore whether we can predict the arrow of time in a temporal sequence, i.e. if it is possible to tell whether a video is running forwards or backwards. We develop a framework using computer vision tools and machine learning to carry out this first part of the project. We then focus on the important types of motion features for predicting the arrow of time. The second part of the project aims at localizing them in the image sequences by exhibiting an influence map for the decision.

1. Introduction

For decades physicist have tried to explain the notion of time. Even if this question seems simple, numerous physicists spend all their lives trying to understand the mystery of time. At the beginning of the twentieth century, Boltzmann spent a great deal of effort in his final years defending his theories on the H-Theorem (early demonstration of the second law of thermodynamics). It was countered by Loschmidt's paradox: how it is possible that there could be a thermodynamic arrow of time given time-symmetric fundamental laws, since time-symmetry implies that for any process compatible with these fundamental laws, a reversed version that looks exactly like a film of the first process played backwards would be equally compatible with the same fundamental laws. And would even be equally probable if one were to pick the system's initial state randomly from the phase space of all possible states for that system. Confronted with this paradox, Boltzmann committed suicide a few years later when he failed to solve the paradox either mathematically or philosophically. Nowadays we know from chaos theory and the fluctuation theorem that their exist an arrow of thermodynamics and our goal will be to assay whether and how the direction of time manifests itself visually. More precisely, we are looking at the reversibility of time in videos. We will investigate, for example, if certain motions can be time-reversed without breaking physics' laws. A trivial case would be an harmonic motions without damping.

2. Extracting motion in image sequences

Based on [3] and their data-set (available at http://www.robots.ox.ac.uk/data/arrow/), we followed a typical framework to compute the motion in images sequence using an estimation of the optical flow. Several methods have arisen to achieve this calculation and we tested both traditional and Convolutional Neural Network (CNN)-based methods.

2.1. CNN for optical flow estimation

Our first attempt to estimate the optical flow was using a pre-trained CNN. We used the codes provided in [4] and followed their guidelines to estimate the optical flow but we never achieved satisfying results. The main problem was the output size, which was too small to catch the motion of small parts (e.g. splashing water) in the videos. The second problem was that the CNN required fine tuning for each videos and therefore we could not obtain a universal framework. The CNN was pre-trained using cartoon-like videos and we think that this type of training might not be fully adapted for real videos. Our next attempt at estimating the optical flow was with a more traditional approach.

2.2. Lucas-Kanade DoG method for optical flow estimation

The Lucas-Kanade derivative of Gaussian method [2] takes two input images and divides them into smaller sections and assumes a constant velocity in each section. Then, it performs a weighted least-square fit of the optical flow constraint equation to a constant model in each section. In order to reduce the noise in the optical flow estimation, we introduced a threshold to put to zero the small velocity values. We thus obtained a sparse matrix containing a vector (V_x, V_y) in each pixel.

2.3. Construction of higher representation features

At this step, supposing that the optical flow is correctly estimated and that the images contain $N = n \times m$ pixels, we have a $2N$ dimensional vector with which we want to tell whether it belongs to a "forward class" or to a "backward class". The curse of dimensionality precludes us from

predicting directly the arrow of time in such a high dimensional space. We therefore need to extract some higher representation features from the optical flow to hope achieving accurate predictions of the arrow of time.

2.4. CNN for extracting features

We used a pre-trained CNN for image classification for this task (matconvnet-vgg-f). As it was trained on RGB images it requires a 3D tensor in input and we decided to introduce some redundancy to create the third dimension:

- The first $n \times m$ matrix is V_x
- The second $n \times m$ matrix is V_y
- The third $n \times m$ matrix is $\theta = arctan\left(\frac{V_y}{V_x}\right)$

The input images size required for using the CNN was $224 \times 224 \times 3$. Our videos were mostly taken in full HD (1920×1080) and we performed a bilinear downsampling, followed by a rescaling (RGB images are ranging in [0,255]) and a normalization (provided in the CNN structure) before feeding the CNN with our artificial images.
For each $224 \times 224 \times 3$ input we only kept one of the fully connected neural network layer (fc7 and fc8), which is a 4096 dimensional vector. For a given video composed by N images, we computed $N - 1$ optical flows and extracted $N - 1$ features vectors. We concatenated these vectors in an unique matrix for further processing with a SVM.

3. Predicting the arrow of time

In the aim of predicting the arrow of time and evaluating our method we followed the same procedure as in [3]. The dataset was divided into 60 testing videos and 120 training videos, in three different ways such that each video appeared as a testing video exactly once and training exactly twice. The backwards-to-forwards video ratios for the three test sets were 9:51, 8:52 and 8:52. The evaluation measure is the proportion of the testing videos on which the method could correctly predict the time-direction.

3.1. Separating motion features with a linear support vector machine (SVM)

In order to distinguish forward and backward motion features we trained a linear SVM. Each optical flow features are used for the training step with their corresponding label. For the testing step, we predict the arrow of time for each optical flow feature vector and proceed to a vote for the final decision: if the number of forward votes are preponderant, then the video is predicted to be forward. We also tried to simply calculate the mean response for a video and the results were quite the same.

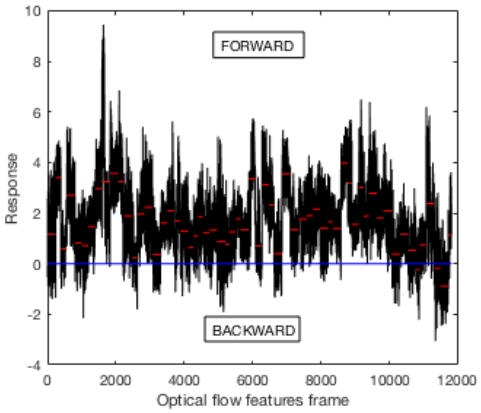


Figure 1. Black: SVM response for each optical flow feature vector - Red: Mean response for a given video - Blue: Boundary between forward and backward classification

Using a cross-validation we obtained the best results for $C = 10$ and with L2 normalization. Using the same protocol as in [3] we obtained the following valid prediction for the arrow of time:

	set 1	set 2	set 3
[3]	75%	90%	77%
fc8 - L2 norm.	83%	92%	88%
fc8 - No norm.	82%	92%	87%
fc7 - L2 norm.	85%	90%	88%

Our method differs from [3] by the fact that we use CNN to extract motion features instead of traditional features (histograms SIFT-like). Using CNN led to an improvement in the prediction of the arrow of time. In order to understand why our method works well we carried out some further experiments.

3.2. Understanding which motions are important for time's arrow prediction

We reversed the 155 forward videos from the data-set. We did the same computation: estimate the optical flow with Lucas-Kanade DoG method and then extract motion features with a CNN. We then computed the differences for each motion vector. In order to remove the noise we applied a threshold on the differences (if the difference is not high enough, it is put to zero). We then summed the differences along both the neurones axis and the videos axis, we then obtained the following figure.

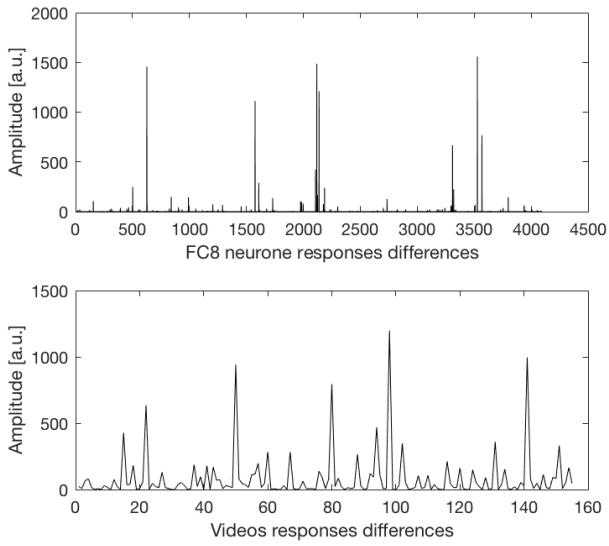


Figure 2. Up: Neurones which are important on FC8 layer for predicting the arrow of time - Down: Videos responses differences between forward and backward

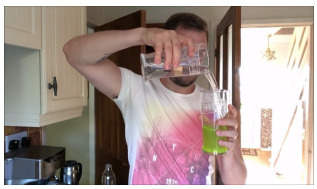
We can see that only a few neurones are specialised into detecting the arrow of time. We can also see that for some videos it is easier to predict the arrow of time. For example, the two videos that present the highest differences in forward versus backward motions are a video of the birth of a jellyfish and a video of the breaking of an iceberg. On the contrary, one of the most difficult video for time direction prediction is a flag moving due to the wind. Locally, the motion of the flag is harmonic and it is therefore difficult to predict, even for a human, the time direction.

Figure 3. Breaking of an iceberg: easy to predict

Figure 4. Flag in wind: hard to predict

3.3. Test on YouTube videos

We run our method on forward and backward videos downloaded from YouTube, see examples below.



We extracted 200 frames for each videos, calculated the optial flow using the Lucas-Kanade DoG method and then extracted feature vectors with CNN (matconvnet-vgg-f, fc8 layer with L2 normalization). We trained a linear SVM with the whole set (180 videos) before testing on YouTube videos. We were able to correctly predict the arrow of time for 4 videos out of 5. The method seems therefore to correctly predict the time direction in image sequences (random responses would lead to 50% accuracy).

4. Localizing key motions in videos for time direction prediction

4.1. Motivation and setup

Once achieved a satisfactory prediction accuracy for time direction prediction, a natural extension is to understand what the system has learnt. This section is devoted to understanding which types of motions the network used for prediction.

In order to do this, we drew inspiration from [1], which proposed a framework for building class activation maps (CAM), a representation that exposes the implicit attention of CNNs on an image. The first key observation made by the authors was that removing the fully connected layers at the end of the network enabled it to keep its localization ability. And secondly, that average pooling was a better fitted criteria than max pooling for the task of precisely activating regions. From the VGGnet, we performed the following steps:

- Remove the fully-connected layers
- Perform global average pooling on the convolutional feature maps
- Add a fully connected layer to produce the categorical output (here forward or backward)
- CAM: projecting back the weights on to the convolutional feature maps

Initializing the network using the pretrained weights from [1], we re-trained this modified network on the optical flows obtained as described above, keeping 20% of the data for testing. Once the class activation maps were obtained on optical flow virtual images, we projected these maps back on the sequences of videos frames for easier interpretation.

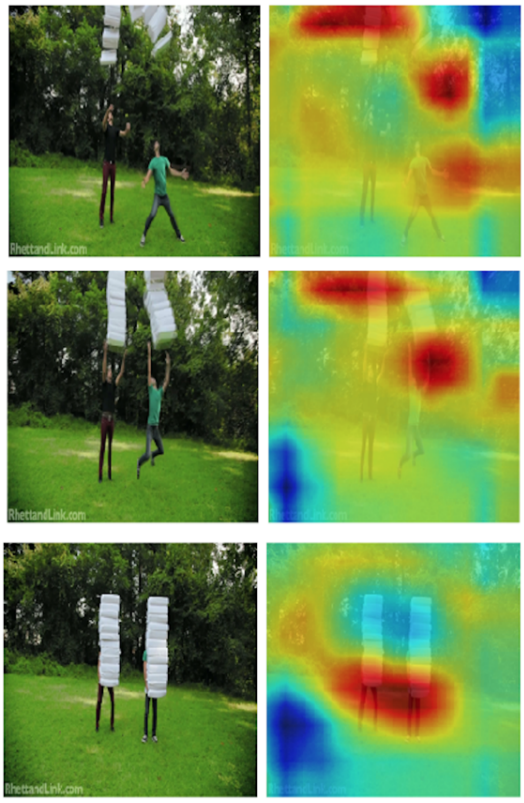


Figure 5. Backward flying pillows activation map

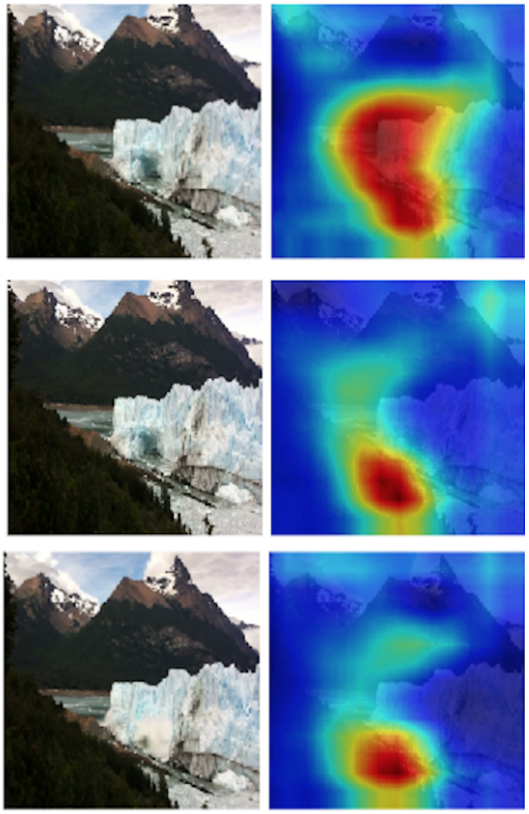


Figure 6. Forward breaking of an iceberg activation map

4.2. Results and discussion

This modified network suffered from a significant drop in classification results. Indeed, although the accuracy on the test set was of 86%, close to previous results, the recall rate for backward videos, which is more adapted to this skewed classification problem, dropped to 36%. As noted by the authors from from [1], this is to be expected, as the fully connected layers are known to highly increase classification results.
Observing the class activation maps generated, we see that the focus on the network (i.e. the activated regions) is on the "moving" parts of successive frames:

- In Figure 5 from a backward video, the focus is first to the stack of pillows in the air (and not on the persons), then to the bent knees.

- In Figure 6 from a forward video, the focus is first on the iceberg falling, then on the water.

Perhaps even more importantly, it seems that indeed, the network has learnt to discard harmonic types of motions and focus on diverging or converging motions. This is obvious in Figure 7, from a forward video of a vintage steam engine. Indeed, the regions containing the piston and the wheel, with harmonic movements are discarded. Whereas the region containing the steam (undetectable in the small figures here), with a clear divergent movement, (and the only part of the video from which it is possible to infer to array of time) is the only one activated.

5. Conclusion

In this work we applied both recent techniques such as CNN and not so recent ones such as Lucas Kanade's optical flow to predict the arrow of time in videos. We yielded a high accuracy (90%), and on broad types of videos and movements. Moreover, wrong predictions seem to be the most difficult videos to predict even for humans (harmonic motion without damping). We then turned to the problem of understanding which regions and which motions are key to predict the arrow of time. We were able to verify that the network focuses on "moving" regions, and on non-harmonic movements more specifically. Some possible improvements may use multi-resolution in time, as time was only captured here by optical flows calculated on successive frames.

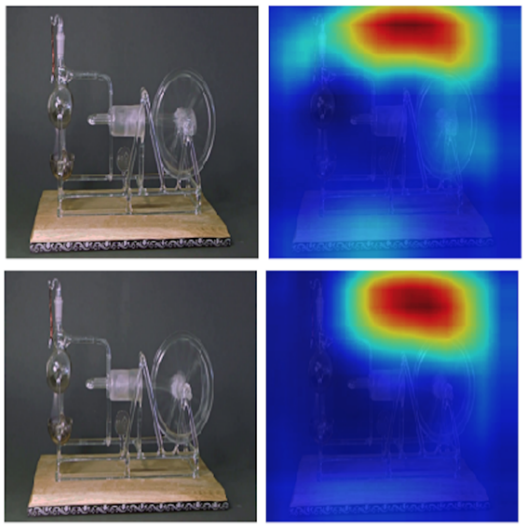


Figure 7. Forward steam engine activation map

References

[1] A. L. A. O. A. T. B Zhou, A Khosla. Learning deep features for discriminative localization. In *CVPR*, 2016.

[2] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[3] L. C. Pickup, Z. Pan, D. Wei, Y. Shih, C. Zhang, A. Zisserman, B. Schölkopf, and W. T. Freeman. Seeing the arrow of time. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[4] D. Teney and M. Hebert. Learning to extract motion from videos in convolutional neural networks. *CoRR*, abs/1601.07532, 2016.