**The Open University of Israel**

**Department of Mathematics and Computer Science**

**Identification of feeding strikes by larval fish from continuous high-speed digital video**

Thesis submitted as partial fulfillment of the requirements

towards an M.Sc. degree in Computer Science

The Open University of Israel

Computer Science Division

By

**Eyal Shamur**

Prepared under the supervision of Dr. Tal Hassner

Dec 2015

**Contents**

[**Abstract** …………………………………………………………………………..6](#_Toc439799472)

[1. **Introduction** 7](#_Toc439799473)

[1.1 Background 7](#_Toc439799474)

[1.2 Larvae’s feeding Identification problem 7](#_Toc439799475)

[1.3 Thesis object 8](#_Toc439799476)

[2. **Previous work** 8](#_Toc439799477)

[3. **Imaging system for digital video recording** 10](#_Toc439799478)

[3.1 Model organisms 10](#_Toc439799479)

[3.2 Experimental set up 11](#_Toc439799480)

[3.3 Manual identification of feeding strikes for a ground-true data 12](#_Toc439799481)

[4. **Feeding event detection by classification** 12](#_Toc439799482)

[4.1 Pipeline overview 12](#_Toc439799483)

[4.1.1 Video pre-processing and fish localization 14](#_Toc439799484)

[4.1.2 Rotation (pose) normalization and mouth detection 15](#_Toc439799485)

[4.1.3 Video clip extraction 17](#_Toc439799486)

[4.1.4 Video representations 17](#_Toc439799487)

[4.1.5 Classification 21](#_Toc439799488)

[5. **Experimental results** 22](#_Toc439799489)

[*5.1 Classification tests* 23](#_Toc439799490)

[5.1.1 Classification benchmark-A 23](#_Toc439799491)

[5.1.2 Classification benchmark-B 25](#_Toc439799492)

[*5.2 Detection tests* 27](#_Toc439799493)

[5.2.1 Detection test procedure 27](#_Toc439799494)

[5.2.2 Detection results 27](#_Toc439799495)

[6. **Summary and future work** 30](#_Toc439799496)

[7. **References** 32](#_Toc439799497)

**List of figures**

[Figure 1: The system overlapping scheme 13](#_Toc439800065)

[Figure 2: Five main blocks of the classification algorithm and their outputs. 13](#_Toc439800066)

[Figure 3: Fish detection 14](#_Toc439800067)

[Figure 4: Pose normalization and mouth detection of larval fish 16](#_Toc439800068)

[Figure 5: Example of a pose-normalized video volume of a feeding fish 17](#_Toc439800069)

[Figure 6: MIP encoding is based on comparing two SSD scores 19](#_Toc439800070)

[Figure 7: Illustration of the dense trajectory description 20](#_Toc439800071)

[Figure 8: ROC for all tested method on classification benchmark-A 25](#_Toc439800072)

[Figure 9: ROC for all tested methods on classification benchmark-B 26](#_Toc439800073)

**List of tables**

[Table 1: Life-history traits for species used in the study. 11](#_Toc433826714)

[Table 2: Run-time performance 22](#_Toc433826715)

[Table 3: Classification benchmark-A results. 24](#_Toc433826716)

[Table 4: Classification benchmark-B results. 26](#_Toc433826717)

[Table 5: Detection results on a video of Hemichromis bimaculatus. (Database A) 28](#_Toc433826718)

[Table 6: Detection results on a video of Sparus aurata.(Database B) 29](#_Toc433826719)

**Acknowledgements**

I wish to thank my thesis supervisor, Dr. Tal Hassner, for his valuable guidance, ideas and helpful remarks throughout the thesis. His assistance, attention to details, hard work and great ideas enriched my knowledge and made this thesis possible.

I wish also to thank Dr. Roi Holzman and his group- from the Department of Zoology, Faculty of life Science, Tel Aviv University and the Inter-University Institute for Marine Science in Eilat (IUI). Dr. Holzman provides the biologic knowledge and background needed for this research. Dr Holzman and his research assistants – Miri Zilka, Alex Liberzon and Victor China, deployed the camera setup for the video recording and manually analyzed the videos to achieve a reliable ground-truth data.

Special thanks to Dalia, my wife that was the woman behind the scene. Her unconditioned support and sacrifice along those long nights allowed me to complete and present this work.

# Abstract

Using videography to extract quantitative data on animal movement and kinematics is a major tool in biomechanics. Advanced recording technologies now enable acquisition of long video sequences in which events of interest are sparse and unpredictable. While such events may be ecologically important, analysis of sparse data can be extremely time-consuming, limiting the ability to study their effect on animal performance and fitness. Using long videos of foraging fish larvae, we provide a framework for automated detection of prey acquisition strikes, a behavior that is infrequent yet critical for larval survival. We compared the performance of four video descriptors and their combinations against manually identified feeding events. For our data, the best single descriptor provided classification accuracy of 95-77%, and detection accuracy of 98-88%, depending on fish species and size. Using a combination of descriptors improved the accuracy of classification by ~2%, but did not improve detection accuracy. Our results indicates that the effort required by an expert to manually label videos can be reduced to examining only the potential feeding detections to filter false detections. Thus, using automated descriptors reduced the amount of work needed to identify events of interest from weeks to hours, enabling the assembly of large, unbiased dataset of ecologically relevant behaviors.

# Introduction

## 1.1 Background

Quantitative analysis of animal movements is a major tool in understanding the relationship between animal form and function, and how animals perform tasks that affect their chances of survival [1]. This discipline benefited greatly with the evolving of digital high-speed videography. Because of practical limitation, such as data analysis, which is an exhaustive, manually operated time-consuming task, analysis is often focused on short video clips, usually <1 second. Events of interest such as the moment of animals while jumping, landing, or striking prey are captured on video by manually triggering the camera at the right time, and saving the relevant range within each video sequence. This way of acquiring data is suitable for events that can be either easily identified in real time, easy to induce, or are repetitive and frequent. However, for events that do not adhere to these criteria or that are unpredictable in space and time, manual triggering and saving short clips limit the possible scope of research. One such example is suction feeding by larval fish.

## 1.2 Larvae’s feeding Identification problem

Systematic observations of larval feeding attempts have proven critical for understanding of the feeding process in order to prevent larval starvation and mortality [2]. However, their implementation was highly ineffective and required considerable effort, limiting its widespread application in larval fish research.

Body length of a hatching larva is a few millimetres, and its mouth is as small as 100 μm in diameter. The high magnification optics required leads to a small depth-of-field and limited visualized area. Fast-cruising larvae remain in the visualized area for only a few seconds. A low feeding rate (especially in the first days after hatching) results in a scarcity of feeding attempts in the visualized area [3]. Similar to adults, prey capture in larvae takes a few tens of millisecond [2] [3] [4] easily missed by the naked eye or conventional video.

Using continuous high speed filming can mitigate some of these shortcomings by providing good spatial and temporal resolution while integrating over several minutes of feeding to increase the probability of observing prey-capturing strike. However, strikes have to be identified by observing the movies ~ 30 – 100 times slower than the recorded speed, a time-consuming task. For example, biologics estimate data acquisition rate as 0.8-3 strikes/hr, depending on larval age when using traditional, burst-type high-speed cameras.

## 1.3 Thesis object

Our goal was to solve the Larva’s feeding identification problem by developing an automated computer vision based method to characterize larval feeding easily in a non-intrusive, quantitative, and objective way. Specifically, we set out to detect prey-capturing strikes from continuous high speed movies of larval fishes. This procedure provides an unbiased, high-throughput method to measure feeding rates, feeding success, prey selectivity, and handling time, as well as swimming speed and strike kinematics.

In addition to solving the Larva’s feeding identification problem, this work provides a benchmark contains 300 clips of larva’s feeding strikes as positive examples, and 300 clips of larva’s non-feeding activities as negative examples. With this benchmark, researches will be able to measure and compare future methods against the current suggested one, while seeking for better performance.

# Previous work

Larva’s feeding may be considered a particular type of action. Larva’s feeding detection is therefore a particular problem within the greater problem of action recognition.   
Action recognition is a central theme in Computer Vision. Over the years action recognition methods have been designed to employ information ranging from high level shape representation to low-level appearance and motion cues. Several early attempts relying on high-level information, include explicit models of bodies in motion [5] , silhouettes [6] , 3D volume [7] or using bank of action templates [8]. In recent years, however, three general low-level representation schemes have been central in action recognition systems. These are the local descriptors, optic flow, and dynamic-texture based representation.

**Local descriptors**. These methods begin by seeking coordinates of space-time interest points (STIP) [9]. The local information around each such point is then represented using one of several existing or adapted feature point descriptors. A video is then represented using, for example, a bag-of-words representation [10].

Some recent examples of such methods include [11] [12]. This approach has proven effective on a number of recent, challenging data sets (e.g., [13] ), yet one of its chief drawbacks is the reliance on a suitable number of STIP detections in each video; videos supplying too few (this is our case - videos of subtle motion) may not provide enough information for recognition. Videos with too much motion (e.g., back-ground, textured motion such as waves in a swimming pool) may drown any informative cues for recognition.

**Optical-flow based methods**. These methods first estimate the optical-flow between successive frames [14] [15] , sub-volumes of the whole video [16], or surrounding the central motion [17] [18] . Optical-flow, filtered or otherwise, provides a computationally efficient means of capturing the local dynamics in the scene. Aggregated either locally (e.g, [17] ) or for whole video volumes as in [14] . Usually, Optical flow methods require heavy computations, however, Violent Flows (ViF) [19] is a simple approach that is capable to run in real-time by considering how flow vector magnitudes change through time and collecting this information over short frame period. Another efficient method that challenges the computation complexity with high performance is Dense Trajectories [20] with its Motion Boundary Histogram (MBH) descriptor. The trajectories are obtained by tracking densely sampled points using optical flow fields. And can be accelerated by GPU computations [21]. The MBH shows that motion boundaries i.e. spatial gradient of the optical flow fields, encoded along the trajectories significantly outperform state-of-the-art descriptors.

Optical-flow based methods commit early-on to a particular motion estimate at each pixel. Unreliable or wrong flow estimates would therefore provide misleading information to any subsequent processing.

**Dynamic-texture representations**. These methods evolved from techniques originally designed for recognizing textures in 2D images, by extending them to time-varying \dynamic textures" (e.g., [22]). The Local Binary Patterns (LBP) [23], for example, use short binary strings to encode the micro-texture centered around each pixel. A whole 2D image is represented by the frequencies of these binary strings. In [22] [24] the LBP descriptor was extended to 3D video data and successfully applied to facial expression recognition tasks. Another LBP extension to videos is the Local Trinary Patterns (LTP) descriptor of [25] . To compute a pixel's LTP code, the 2D patch centered on it is compared with 2D patches uniformly distributed on two circles, both centered on its spatial coordinates: one in the previous frame, and one in the succeeding frame. Three values are used to represent whether the central patch is more similar to one in the proceeding frame, the succeeding frame or neither one. A string of such values represents the similarities computed for the central patch with the patches lying on its two corresponding circles. A video is partitioned into a regular grid of non-overlapping cells and the frequencies of the LTP codes in each cell are then concatenated to represent the entire video.

The Motion Interchange Pattern (MIP)[26] descriptor extends the LTP descriptor to 8 directions, and treats each direction separately. To decouple static image edges from motion edges MIP incorporate a suppression mechanism, and to overcome camera motion it employs a motion compensation mechanism. A bag-of -words approach is then used for each direction to pool information from the entire video clip. All direction bag-of-words are than concatenated to form a descriptor.

In this work we check all of the three schemes by using at least one primary method for a scheme. MIP, STIP, VIF and MBH were used as described in section 4.1.4.

# Imaging system for digital video recording

## 3.1 Model organisms

We focused on three fish species: 13-23 DPH (days post-hatching) *Sparus aurata* Linnaeus, 1758 (gilthead sea bream; Sparidae, Perciformes, Actinoperygii), 14-16 DPH *Amatitlania nigrofasciata* Günther, 1867 (Cichlidae, Perciformes Actinopterygii), and 8-15 DPH *Hemichromis bimaculatus* Gill, 1862 (Cichlidae, Perciformes Actinopterygii). *S. aurata* is a marine fish of high commercial importance, commonly grown in fisheries, while the two cichlid species are freshwater fish that are grown in the pet trade. Sparus aurate has life history that is characteristic of pelagic and coastal fishes, while the cichlids provide parental care to their offspring. Thus, the cichlid larvae hatch at a much larger size and as more developed larvae (Table 1).

|  |  |  |  |
| --- | --- | --- | --- |
|  | *Sparus aurata* | *Amatitlania nigrofasciata* | *Hemichromis bimaculatus* |
| Egg diameter at hatching [mm] | ~1 | ~1.3 | ~1.3 |
| Length hatched larvae [mm] | 3.5 | 5.0 | 4.9 |
| Age at filming [DPH] | 13, 23 | 8, 11, 15 | 8, 14, 16 |
| Length at filming [mm] | 4.5, 6.5 | 5.6-6.1 | 5.5-5.9 |
| Number of events used for classification | 300 | 300 | |

Table : Life-history traits for species used in the study.

## 3.2 Experimental set up

During experiments, larvae were placed in small rectangular experimental chamber (26 x 76 x 5 mm). Depending on fish age and size, 5-20 larvae were placed in the experimental chamber and were allowed several minutes to acclimate before video-recording began. Larval density was adjusted so that at least one larva would be present at the field of view through most of the imaging period. Typical feeding sessions lasted 5-10 minutes. Rotifers (*Brachionus rotundiformis*; ~160 μm in length) were used as prey for all fish species as they are wildly used as the standard first-feeding food in the mariculture industry.

Visualization of *Sparus aurata* larvae was done using a continuous hi-speed digital video system (Vieworks VC-4MC-M/C180, operating at 240 frames per second with resolution of 2048×1024. The camera was connected to a PC, and controlled by Streampix 5 video acquisition software (Norpix, Montréal, Canada). A 25 mm *f* /1.4 C-mount lens (Avenir CCTV lens, Japan) was mounted on a 8 mm extension tube, providing a field of view of 15 x 28 x 3 mm (height , width and depth, respectively) at *f*=5.6. We used backlit illumination, using an array of 16 white LEDs (~280 lumen) with a white plastic diffuser. To increase computation efficiency, original videos were rescaled to 1024x512 pixels per frame. This size was empirically determined to accelerate computation without having an impact on the final accuracy.

## 3.3 Manual identification of feeding strikes for a ground-true data

Following recording, films were played back at reduced speed (15 fps) in order to manually identify feeding attempts. Overall, we obtained 300 feeding events for the three species used in this study: *s. aurata* (23 DPH and 13 DPH) and two cichlid species (*Amatitlania nigrofasciata, Hemichromis bimaculatus* ). These feeding events used as ground truth for our tested methods, and used as positive examples in our benchmark.

# Feeding event detection by classification

## 4.1 Pipeline overview

A block diagram of the feeding event detection process is provided in Figure 2. Key to its design is the decoupling of fish detection and pose normalization, from the representation of local spatio-temporal regions, and their classification as either feeding or non-feeding events. We begin by preprocessing the entire video in order to detect individual fish, discriminating between them and their background and other noise and artifacts in the video (step a in Figure 2, detailed in Section 4.1.1). Following this step, each fish is analyzed to determine the location of its mouth and rotated to a roughly horizontal position to provide rotation (pose) invariance (step b, Section 4.1.2). Small spatio-temporal volumes (“clips”) around each mouth are extracted (step c, Section 4.1.3) and represented using robust video descriptors (step d, Section 4.1.4). Finally, classification to feeding / non-feeding is performed using a radial basis function (RBF) support vector machine (SVM) classifier (step e, Section 4.1.5).

Due to the high ratio between frame rate (240fps) and the duration of feeding attempts (usually < 60 ms), the classification processing did not need to be applied at every frame to reliably identify feeding attempts. We therefore empirically set the system to process 21 frame volumes only every 10th frame for *A. nigrofasciata* and *H. bimaculatus* or 41 frame volumes only every 20th frame for the slower feeding *S. aurata*. The duration of our clips is twice as long as the gap between the center frames, ensuring that no frame is left unprocessed; The larva is monitored for the entire duration in the field of view; every potential feeding event is captured by at least two clips, as the extracted volumes overlaps. 11-frame-overlapping and 21-frame-overlapping accordingly as demonstrated by Figure 1.

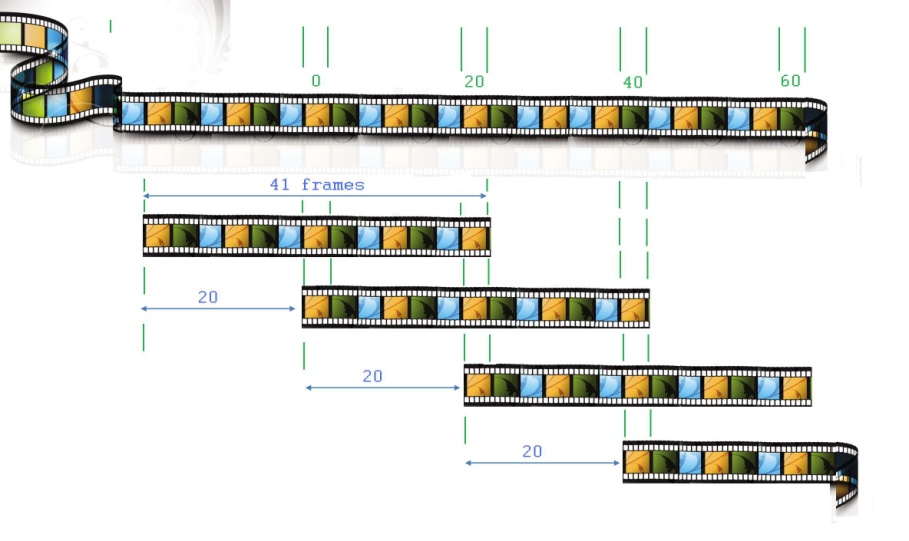


Figure 1: The system overlapping scheme

The system processes 41 frame volumes only every 20th frame for the slower feeding S. aurata.

In the following sections we describe each of these steps in detail.

|  |
| --- |
| e) Classification  d) Visual descriptor  c) Video clip extraction  a) Video preprocessing  b) Fish rotation & mouth detection  Decision – feeding fish or non-feeding fish  Fish blobs  Fish mouth location  Video clip descriptor  Fish mouth video clip |

Figure : Five main blocks of the classification algorithm and their outputs.

## 4.1.1 Video pre-processing and fish localization

In our data, typical video frames contained measurement noise, resulting from floating food particles, light/shadow specks, reflections on the water’s surface, and dirt on the bottom of the chamber. Figure 3a provides an example frame from our camera setup. Our processing begins by attempting to remove much of this clutter. We first apply a standard image segmentation technique [27] followed by binary separation of the video to foreground/background pixels, in order to separate the background from noise and fish blobs (Figure 3 b)

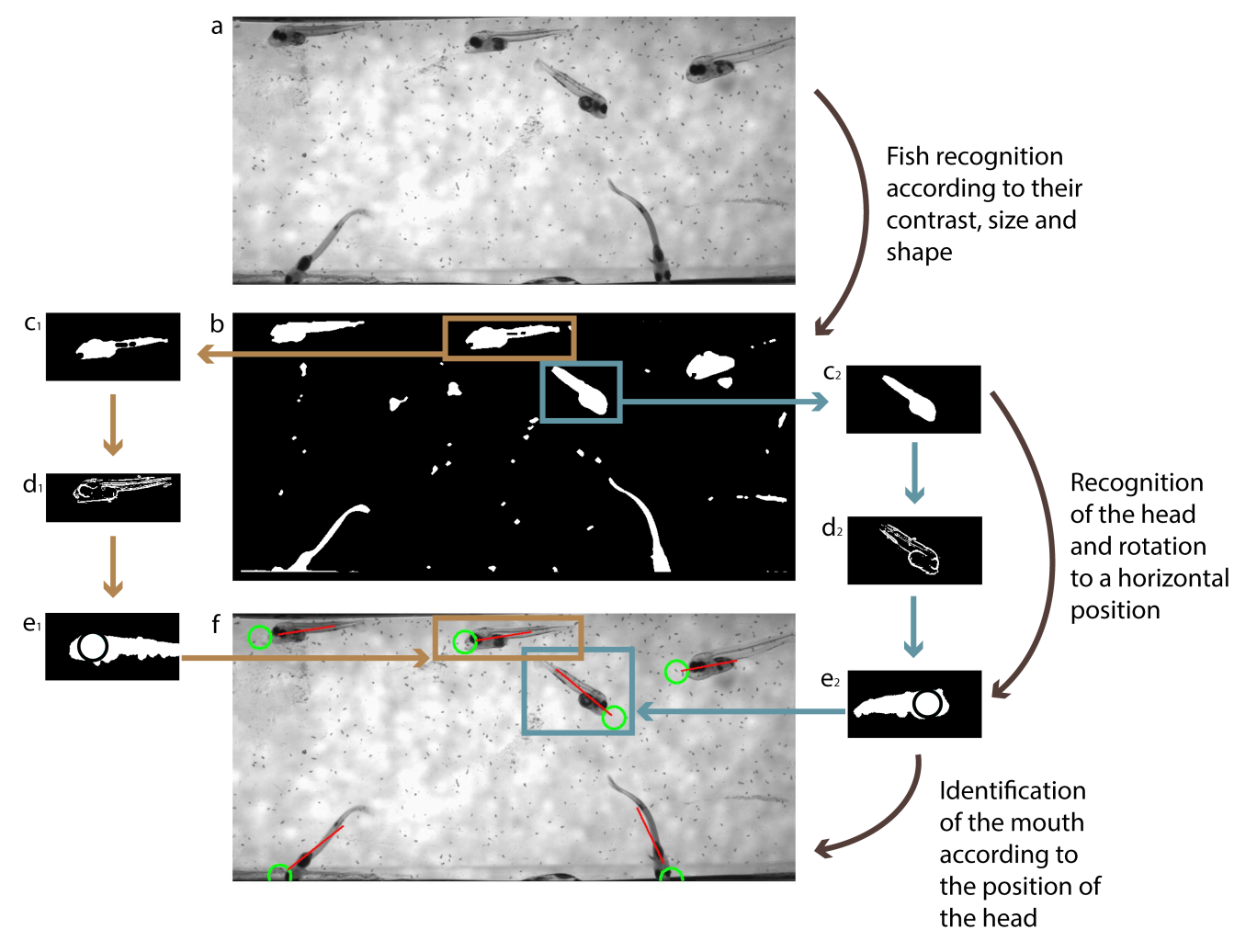


Figure : Fish detection

Video processing to identify fish and determine mouth location (stages a-b in Fig 2). a) an image is selected from the video (here, 23 DPH S. aurata). b) binary separation of the foreground and background is followed by blob extraction (blue and brown insets in b). c) blobs qualified by an eigenvalue ratio test (having appropriate length/width ratios) are maintained, while small blobs are removed. d) gradient analysis is used to identify textured elements (fish) from non-textured ones (noise). e) pose normalization is applied to the blobs. The fish head is located by examining the radius of the maximum bounded circle. f) the main axis of the fish body and the head are visualized, and projected on the original image: green circles pointed to fish mouths, and red lines represent fish bodies’ main (long) axis.

The fish species in our videos, all had similar size and length-to-height (maximal dorso-lateral distance) ratio. We therefore remove foreground blobs having less than a set threshold number of pixels or having more than. Non fish-shaped blobs were then removed by considering the ratio between the two eigenvalues and of each foreground segment. A blob was removed if the following condition does not hold:

The value for was set to 350 pixels for Sparus 13dph and 800 for other species. The value for was set to 10000 pixels. and were set to 100 and 1, reflecting overly elongated segments and near circular shapes. These values were determined empirically, and were not changed throughout our experiments.

The process above eliminates most of the non-fish foreground blobs, but some blobs may still share the same size or shape as fish. These are identified by considering the texture within each blob; blobs produced by noise typically present flat appearances compared to the textured fish bodies. Specifically, we evaluate the following expression for each foreground blob:

Where

Here, , where is the horizontal image gradient and the vertical gradient, both at the *i*’th pixel of *k*’th blob and both approximated using standard 3x3 Sobel filters. The values for the two thresholds and where set to 120 and 140, and used throughout our experiments. These steps are visualized in Figure 3c and Figure 3d.

## 4.1.2 Rotation (pose) normalization and mouth detection

As fish swim freely in their tank, their heads may be oriented in any possible direction. This is quite different from standard action recognition applications where actions are typically performed oriented in the same manner: a video of a human actor walking would typically have the motion of the legs appearing at the bottom of the frame, below the rest of the body. Representations used to capture and discriminate between human actions are therefore not designed to be invariant to the rotational differences exhibited by our fish. Here, this invariance is introduced prior to feature extraction by rotating all fish-head spatio-temporal blobs to a canonical position, in a manner similar to the one employed by low-level descriptors such as SIFT [28]

Specifically, at the particular larva development stage considered here, the head is substantially bigger than any other part of its anatomy. The head can therefore be detected simply by locating the max-bounded circle of the fish segment and the mouth assumed as the nearest blob end. The spatio-temporal volume around the fish mouth is then rotated to align the X-axis of the entire fish blob with the frame’s horizontal axis, using standard principle component analysis (PCA). Additional invariance to reflection is then introduced by reflecting all spatio-temporal volumes in order to produce horizontally-aligned, right-facing fish.

The two steps of detecting fish mouths and rotating the segments are visualized in Figure 3e and in Figure 4. Figure 3f provides example mouth and axis detection of multiple fish in the same frame.

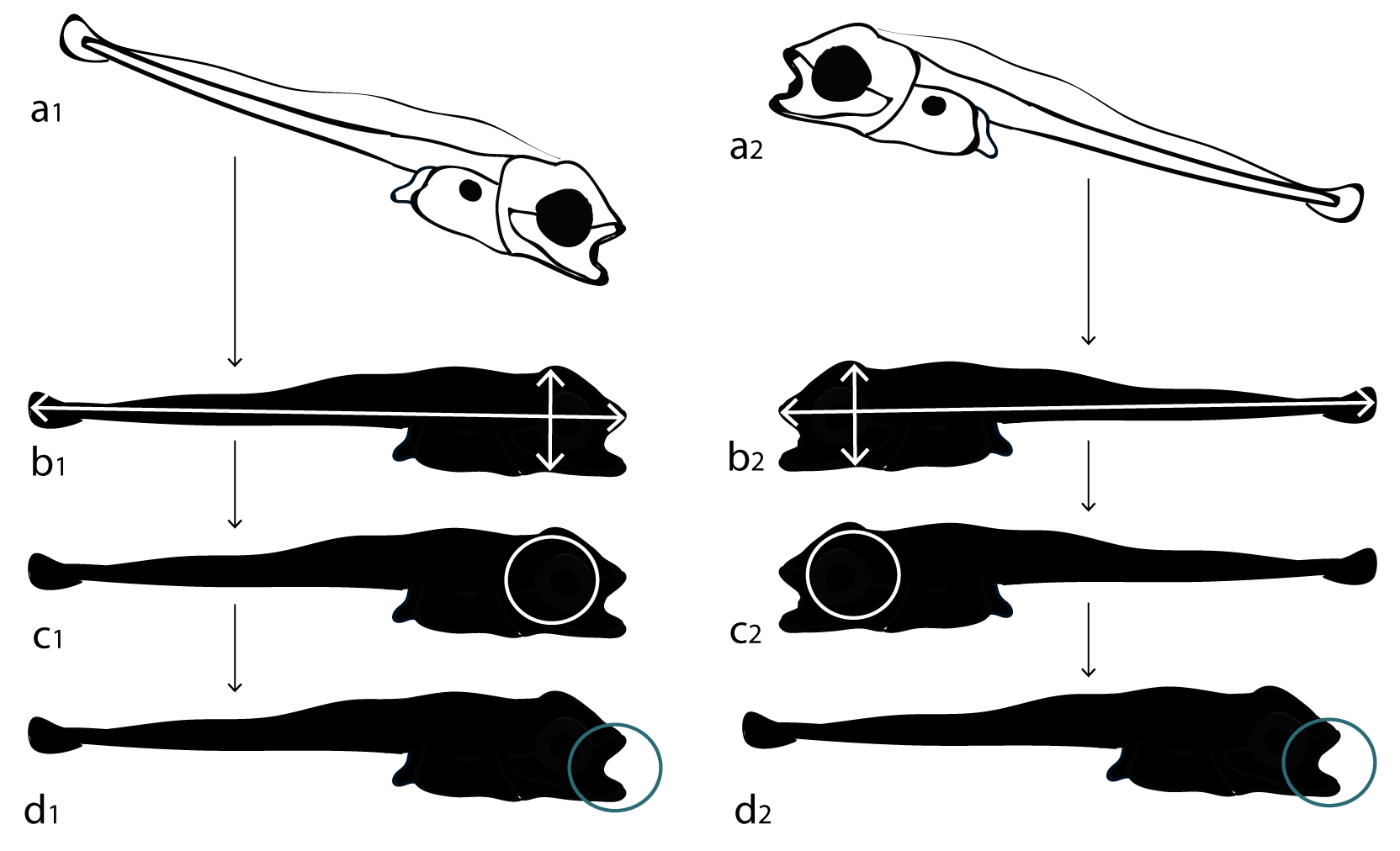


Figure : Pose normalization and mouth detection of larval fish

## 4.1.3 Video clip extraction

The result of this stage of our processing is a defined area around each detected mouth. We extract 121x121 pixels centered on the mouth’s central pixel for 21 frames from the compressed video (for the Amatitlania *nigrofasciata* and *Hemichromis bimaculatus*) or 241x241 pixels for 41 frames from the original hi-res video (for the slower eating *Sparus aurata*). The choice of spatial dimensions allows coverage of entire heads, along with sufficient margins for possible food floating around the fish. Temporal dimension were empirically determined to be long enough to span feeding. Figure 5 depicts frames from an example pose-normalized video volume of a feeding fish.

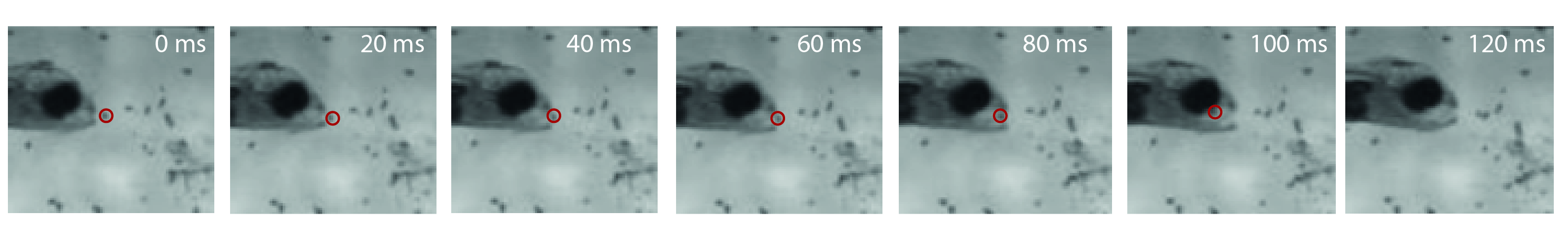


Figure : Example of a pose-normalized video volume of a feeding fish

Extracted spatio-temporal volume in canonical views (horizontal, right-facing views) of a feeding fish. The prey is marked by a red circle, and enters the mouth at 60 ms. The mouth is closed at 120 ms.

## 4.1.4 Video representations

The pose-normalized video volumes produced in the previous step are next converted to robust representations, whose function is to represent actions appearing in videos. These have been designed to capture discriminative information unique to different actions, as well as remain robust to small differences in how each action is performed, the actor performing it, the viewing conditions and more. We experimented with a number of recent video representations, previously shown – see previous work on paragraph 2 - to provide excellent action recognition performance among other descriptors of their kind. Specifically, each pose-normalized volume was encoded using the following action descriptors: (1) The Space Time Interest Points (STIP)[[1]](#footnote-1) of [9]; (2) The Motion Interchange Patterns (MIP)[[2]](#footnote-2) of [26]; (3) TheDense trajectories and Motion Boundary Histogram (MBH)[[3]](#footnote-3) presented in [21]; and (4) the Violent Flows descriptor (ViF)[[4]](#footnote-4) of [19]. The first three have been shown to provide excellent action classification performance on videos of humans performing a wide range of actions. The last was designed specifically for fast detection of violent actions. All four have been shown in the past to be complementary of each other (e.g., [26]). As we later show, combining these representations indeed substantially elevates detection accuracy.

(1) STIP descriptor [9]

Inspired by Harris corners, the idea of STIP is to find spatio-temporal location where a video imagehas significant change in three directions - the two spatial directions and the temporal direction. For a given spatial variance and temporal variance such a point can be found using a second moment matrix integrated over a Gaussian window. If is a Gaussian window on then the second moment matrix M is:



While



and the second moment matrix over a second Gaussian window is:



Our interest points are those having significant eigenvalues  of . On [9] it is shown that positive local maxima of the corner function *H* correspond to these interest points with high variation of image values along spatial and the temporal directions.



Having found interest points, STIP descriptor is then described as BoW of HoG, HoF or concatenated HogHof.

(2) MIP descriptor [26]

* Inspired by the LTP [29] descriptor, MIP encodes every pixel on every frame by eight strings of eight trinary (-1,0 or1) digits each. Each digit compare the compatibility of two motions with the local patch similarity pattern: one motion in a specific direction from the previous frame to the current frame, and one motion in a different direction from the current frame to the next one. A digit value of -1 indicates that the former motion is more likely, 1 indicates that the latter is more likely. A value of 0 indicates that both are compatible in approximately the same degree. Going in 8 directions, MIP provides a complete characterization of the change from one motion to the next.

The encoding is based on comparing two SSD scores computed between three patches from three consecutive frames, see Figure 6.

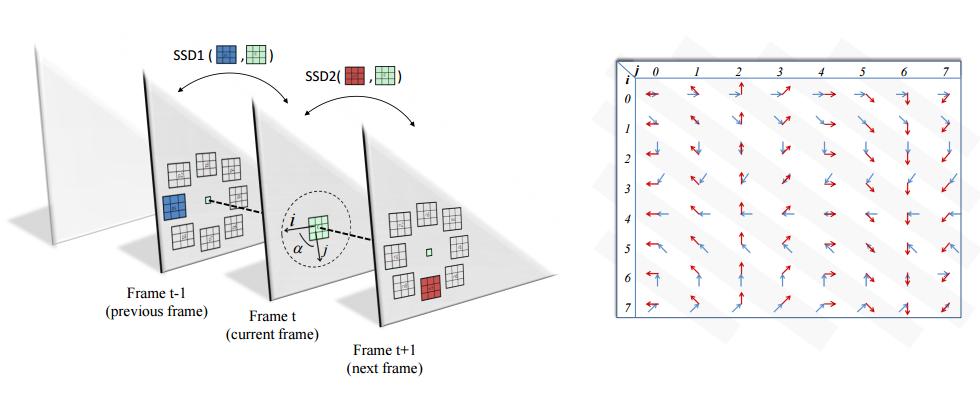


Figure 6: MIP encoding is based on comparing two SSD scores

computed between three patches from three consecutive frames. Relative to the location of the patch in the current frame, the location of the patch in the previous (next) frame is said to be in direction i

Digit values are determined by



Each pixel in the video is encoded by one 8-trit string per channel (i.e. direction). As in LTP the positive and negative parts of the strings separately obtaining 2 UINT8 per pixel. The first UINT zeros the -1 digits and the second UINT zeros the 1 digits.  
These 16 values represent the complete motion interchange pattern for that pixel. For eac channel the frequencies of these MIP codes are collected in small 16x16 patches in the image to create 512 dimensional code words. Video representation is done by concatenating cell’s histograms into 512 length descriptor.

(3) MBH descriptor [21]

This approach describes video by dense trajectories. Dense trajectories are obtained by tracking densly sampled points – on a grid spaced W (=5) pixels - using optical flow fields for multiple spatial scales. Tracking is proposed in the corresponding spatial scale over L frames (usually L=15). See Figure 7 . trajectory descriptors are based on its coarse represented by HoG, HoF or MBH over a local neighborhood of NxN pixels (usually N=15). In order to capture the structure information, the trajectory neighborhood is devided into a spatio-temporal grid of size nσ ×nσ ×nτ. Setting N = 32, nσ = 2, nτ = 3.

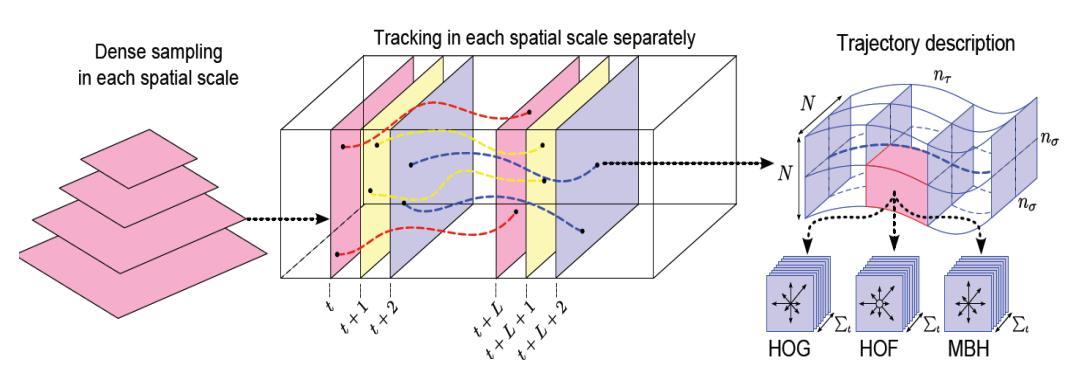


Figure : Illustration of the dense trajectory description

The MBH descriptor separates the optical flow field into its x and y component. Spatial derivatives are computed for each of them and orientation information is quantized into histograms.

MBH descriptor dimension is 96 (i.e., 2 × 2 × 3 × 8) for both MBHx and MBHy.

(4) ViF descriptor [19]

This method that has been proposed for real time detection of breaking violence in scenes, considers statistics of how flow-vector magnitudes change over time. The Violence Flows (ViF) descriptor first estimates the optical flow between pairs of consecutive frames, providing for each pixel a flow vector . This flow vector is matched to the flow vector of a pixel in the previous pair of frames  providing a binary score .



A mean magnitude-change map is then computed by simply averaging these binary values, for each pixel, over all the frames in a video volume A. The ViF descriptor is therefore produced by partitioning b into M × N non-overlapping cells and collecting magnitude change frequencies in each cell separately.



The distribution of magnitude changes in each such cell is represented by a fixed-size histogram. These histograms are then concatenated into a single descriptor vector.

## 4.1.5 Classification

Binary classification of each pose-normalized space time volume as either representing a feeding / non-feeding event is performed by first extracting feature descriptors , where represents STIP, MIP, MBH, or VIF. Each video clip has a set of hundreds of descriptors for each type. All videos together produce more than 50K of descriptors. We adopted the BoW (Bag of Words) paradigm suggested by [10]: The descriptors above where classified into 512 or 128 bins using k-Means, generating a 512 or 128 bin histogram – Bag of Words - for each video clip. Finally, these BoW histograms were classified into feeding class or into non-feeding class using standard support vector machines (SVM)[[5]](#footnote-5) with RBF kernels [4].

SVM was directly applied to discriminate between descriptors extracted from each pose-normalized volume. In addition, we performed tests with combinations of these descriptors. Multiple descriptors were evaluated by stacking SVM classifiers [30] as stacking SVM was proven to outperform the single SVM. Specifically, decision values of SVM classifiers applied separately to each representation were collected in a single vector. These vectors of decision values were then classified using an additional linear-SVM.

# Experimental results

Our tests were conducted on a standard Win7, Intel i7 machine. Table 2 provides a breakdown of the times required for each of the steps in our pipeline. The major bottleneck is evidently the MIP descriptor for which only non-optimizes, MATLAB code exists. As we later show, the accuracy of the two fastest descriptors, MBH and VIF, is nearly as high as the accuracy obtained by combining all descriptors. These two descriptors may therefore be used on their own whenever computational costs must be considered.

|  |  |
| --- | --- |
| Step | Time (sec.) |
| Per-frame | |
| Compression | 0.042 |
| Fish head and mouth detection | 1.07 |
| Per-volume | |
| Pose normalization (rotation and mirroring) | 0.21 |
| STIP encoding\* | 7.35 |
| MIP encoding | 7.01 |
| MBH encoding\* | 1.02 |
| VIF encoding | 4.01 |
| SVM classification | 0.01 |

Table 2: Run-time performance

Break-down of the time required for each of the components of our system. \* All steps of our method were implemented in MATLAB except STIP and MBH encodings and the SVM classification, which were available as (much faster) pre-compiled code

We evaluate the performance of our method using two steps of tests. On the first step we test the classification part which is the core of our identification method, and on the second step we test our overall identification method: (1) First step - Classification tests were conducted in order to learn and evaluate the classification models while trying to classify clips as feeding or as non feeding events . Best models were kept in order to be used later by the detection test procedure as the classification core algorithm. (2) Second step - Detection tests. These tests test and evaluate the whole method. We test the detection correctness of feeding and non-feeding events on the original videos. The detection tests use the models learned previously during classification tests. It must be noted that these models should be learned only once, while they can be used multiple times. In both classification and detection, our tests were applied separately to the faster eating fish, *A. nigrofasciata* and *H. bimaculatus* and to the slower *S. aurata*

# **5.1 Classification tests**

Our classification benchmarks each includes pose-normalized volumes which were extracted using the process described in Figure 2. We measure binary classification rates for eating vs. non-eating events and compare our system’s performance vs. manually labelled ground truth. We note that testing the classification of pre-detected instances in this manner is standard practice in evaluating action recognition systems, particularly when positive events are very rare, as they are here (see [31] for a survey of contemporary action recognition and detection benchmarks).  
Nevertheless, this paper includes also video detection rates, in the next sections.

5.1.1 Classification benchmark-AThis benchmark contains 150 videos of eating events and 150 videos of non-eating events, of *Amatitlania nigrofasciata* and *Hemichromis bimaculatus.* Both species have similar morphology and strike kinematics, and consequently were collectively treated in the same benchmark.   
  
  
We use a leave-one-out, six-fold, cross-validation, test protocol. Each fold contains 50 video-exclusive volumes; that is, a video contributes volumes to only one fold, thereby preventing biases from crossing over from training to testing. In each of the six tests, 250 volumes are used to train the SVM classifiers, and 50 are used for testing. In each test split, half of the volumes portray eating events and half do not.  
Results are reported using mean accuracy (ACC) ± standard error (SE) computed over all six splits. Here, mean accuracy is the average number of times our system predicted an eating vs. non-eating event on our sets of volumes and standard error was measured across the six test splits. We provide also the overall AUC: the area under the receiver operator curve (ROC). Finally, we provide the sensitivity (true positive / positives) and specificity (true negative / negative). – listed above ACC is “accuracy”, ROC is “receiver operator curve”.  
Our results are presented in Table 3 with ROC for all tested methods provided in Figure 8. Evidently, the highest performance was obtained by the combined representation, where MBH alone was responsible for much of the performance (row h). Interestingly, the fastest representations, MBH, obtained nearly the best result (row c), making it an attractive option whenever computational resources are limited.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Specificity | Sensitivity | AUC | ACC ± SE | Descriptor Type |  |
| 70.00 | 69.34 | 0.8076 | 69.66% ± 3.9 | STIP | a |
| 66.67 | 75.34 | 0.9322 | 86.00% ± 2.1 | MIP | b |
| 87.34 | 94.67 | 0.9802 | 91.00% ± 1.1 | MBH | c |
| 78.00 | 71.34 | 0.7783 | 74.67% ± 2.3 | VIF | d |
| 88.00 | 94.00 | 0.9656 | 91.00% ± 1.2 | MBH+VIF | e |
| 86.00 | 94.00 | 0.9731 | 90.00% ± 2.0 | STIP+MIP+MBH | f |
| 88.00 | 96.00 | 0.9725 | 92.00% ± 1.0 | MIP+MBH+VIF | g |
| 89.33 | 96.00 | 0.9724 | 92.67% ± 1.4 | STIP+MIP+MBH+VIF | h |

Table : Classification benchmark-A results.

We provide classification accuracy (ACC) ± standard error (SE), the area under the Receiver operating characteristic curve (AUC), the sensitivity and specificity of each of the tested methods. Shaded row indicates the best result.

# ROC Database-A.jpg

Figure : ROC for all tested method on classification benchmark-A

5.1.2 Classification benchmark-BA separate benchmark was collected for the smaller, slower eating *Sparus aurata* fish. It includes 150 volumes of eating events and 150 non-eating events. Our protocol here is similar to the one used for Benchmark-A, using again six-fold cross validation in which each test involves a training set of 250 clips and test sets on the remaining 50 clips.  
Our results are reported in Table 4 with the ROC provided in Figure 9. The slower eating, larger fish in this benchmark were harder to classify, as each eating event spanned more frames and so produced more subtle differences in the descriptor encodings. This was most evident in the VIF descriptor, originally designed to capture fast, violent actions. Its performance on this set degraded substantially (row d). Here too, the best performance was obtained by a combination of descriptors (row h).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Specificity | Sensitivity | AUC | ACC ± SE | Descriptor Type |  |
| 74.00 | 63.34 | 0.7548 | 68.33% ± 2.3 | STIP | a |
| 66.00 | 66.67 | 0.7675 | 66.33% ± 1.9 | MIP | b |
| 70.00 | 72.00 | 0.7671 | 71.00% ± 2.6 | MBH | c |
| 60.00 | 64.00 | 0.6620 | 62.00% ± 1.1 | VIF | d |
| 70.00 | 70.00 | 0.7745 | 70.00% ± 1.1 | MBH+VIF | e |
| 67.33 | 74.00 | 0.8151 | 70.67% ± 2.1 | STIP+MIP+MBH | f |
| 69.33 | 72.00 | 0.8017 | 70.67% ± 2.3 | MIP+MBH+VIF | g |
| 70.00 | 75.33 | 0.8183 | 72.67% ± 2.1 | STIP+MIP+MBH+VIF | h |

Table 4: Classification benchmark-B results.

We provide classification accuracy (ACC) ± standard error (SE), the area under the Receiver operating characteristic curve (AUC), the sensitivity and specificity of each of the tested methods. Shaded row indicates the best result.

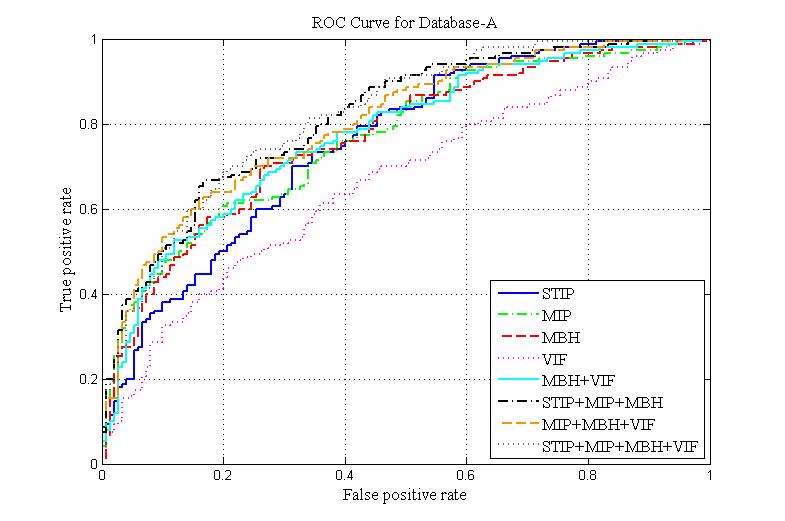
****

Figure : ROC for all tested methods on classification benchmark-B

# 

# **5.2 Detection tests**

## 5.2.1 Detection test procedure

We next measure the rate at which our pipeline correctly detects feeding events in videos. Our tests were performed on a video with 6,000 frames of *Hemichromis bimaculatus* fish, which included 14 manually labeled eating events. Our pipeline decomposed this video into a total of 535 pose-normalized volumes. Separate tests were performed on a video of 4,200 frames capturing *Sparus aurata* fish. Here, only five feeding event were manually labeled, compared to a total of 451 pose-normalized volumes automatically extracted by our system.

In our detection tests we report the following performance measures for each video: True positive (TP) which is the number of times an eating fish was detected as such, true negative (TN) – the number of times a non-eating fish was detected as such -- and accuracy (percent of volumes correctly detected as either eating or non-eating). We provide also the confusion matrices for each test, showing the detection rates (in percentages) of predicted feeding and non feeding events (Pred. feed and Pred. non-feed, respectively) vs. actual labels for each event (Feed and No-feed). Here too, as with our classification tests, we report performance for all descriptors and their combinations.

## 5.2.2 Detection results

Detection performance on a video of *Hemichromis bimaculatus* are provided in Table 5 and performance on a *Sparus aurata* video is provided in Table 6. In both cases, MBH excels, compared to other representations and even representation combinations. These numbers, however, should be considered along with the small total number of feeding events in the video, which implies that small variations in performance may not be statistically significant. Regardless, both tests show that our system has no false positives and only a small rate of false negatives. This is ideal, as it implies that it can reduce the effort required by an expert to label videos to examining only a few predicted feeding detections: no true feeding events are missed and only a negligible number of false detections (false negatives) are left over to examine and manually filter.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Descriptor |  | Confusion Matrix | | TP | TN | Acc |
|  |  |  | Pred. feed | Pred. no-feed |  |  |  |
| a | STIP | Feed | 100.00% | 0.00% | 100% | 66% | 83% |
| No-feed | 34.17% | 65.83% |
| b | MIP | Feed | 92.86% | 7.14% | 93% | 83% | 88% |
| No-feed | 17.23% | 82.77% |
| c | MBH | Feed | 100.00% | 0.00% | 100% | 95% | 98% |
| No-feed | 4.99% | 95.01% |
| d | VIF | Feed | 92.86% | 7.14% | 93% | 70% | 81% |
| No-feed | 30.31% | 69.69% |
|  |  |  |  |  |  |  |  |
|  |  |  |
| e | MBH+VIF | Feed | 100.00% | 0.00% | 100% | 91% | 95% |
| No-feed | 9.02% | 90.98% |
| f | STIP+MIP+MBH | Feed | 100.00% | 0.00% | 100% | 86% | 93% |
| No-feed | 13.51% | 86.49% |
| g | MIP+MBH+VIF | Feed | 100.00% | 0.00% | 100% | 89% | 94% |
| No-feed | 11.37% | 88.63% |
| h | STIP+MIP+MBH+VIF | Feed | 100.00% | 0.00% | 100% | 83% | 92% |
| No-feed | 16.60% | 83.40% |

Table : Detection results on a video of Hemichromis bimaculatus. (Database A)

Each row provides detection performance using a different video representation. Results include the confusion matrix for true vs. predicted feeding and non-feeding events (appearing shaded), the True positive rate (TP), true negative rate (TN) and the accuracy (Acc).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Descriptor |  | Confusion Matrix | | TP | TN | Acc |
|  |  |  | Pred. feed | Pred. no-feed |  |  |  |
| a | STIP | Feed | 100.00% | 0.00% | 100% | 63% | 82% |
| No-feed | 37.00% | 63.00% |
| b | MIP | Feed | 100.00% | 0.00% | 100% | 70% | 85% |
| No-feed | 30.04% | 69.96% |
| c | MBH | Feed | 100.00% | 0.00% | 100% | 75% | 88% |
| No-feed | 24.66% | 75.34% |
| d | VIF | Feed | 100.00% | 0.00% | 100% | 60% | 80% |
| No-feed | 39.69% | 60.31% |
|  |  |  |  |  |  |  |  |
|  |  |  |
| e | MBH+VIF | Feed | 60.00% | 40.00% | 60% | 75% | 68% |
| No-feed | 24.89% | 75.11% |
| f | STIP+MIP+MBH | Feed | 100.00% | 0.00% | 100% | 74% | 87% |
| No-feed | 25.56% | 74.44% |
| g | MIP+MBH+VIF | Feed | 100.00% | 0.00% | 100% | 75% | 88% |
| No-feed | 24.78% | 75.22% |
| h | STIP+MIP+MBH+VIF | Feed | 100.00% | 0.00% | 100% | 74% | 87% |
| No-feed | 25.56% | 74.44% |

Table : Detection results on a video of Sparus aurata.(Database B)

Each row provides detection performance using a different video representation. Results include the confusion matrix for true vs. predicted feeding and non-feeding events (appearing shaded), the True positive rate (TP), true negative rate (TN) and the accuracy (Acc).

# Summary and future work

Visualization of larval feeding is challenging due to size, timescale, and rarity of feeding events at early larval stages. However, visualization is essential for measuring the rate of feeding attempts and failed attempts. Identifying feeding attempts by means of the human eye is a time-consuming process; by automating this process, we will not only ensure objectivity but also enable data acquisition on in a larger scale than ever obtained to date in the field of larval feeding. Automatic software identification of feeding attempts will eliminate the current bottleneck for acquiring data.

We present a novel method that can be used to automatically identify prey acquisition strikes in larval fishes, facilitating the acquisition of large data sets from rare, sparse events. In the case of larval fish, this method can be used to assess feeding rates and success, to determine the fate of food particles during the feeding cycle, and to perform detailed kinematic analysis or prey acquisition strikes, helping to build a better understanding of the factors that control prey acquisition. More generally, this method can be applied to any model system where specific locomotory tasks cannot be easily actuated. This could be especially important in studies of natural behaviors in field conditions, or when considering infrequent events.

We believe that our approach can advance computational work for the modeling of larval feeding, leading to a better understanding of the specific larval failure mechanisms in the feeding process. Our method can be employed in a wide range of studies on larval feeding: the effect of inter- and intra- species competition, food preferences and feeding selectivity, prey escape response, and predator-prey co-evolution. All of these represent some of the enormous potential our approach can offer.

Future researches in this field could improve current results and could expand to wider areas. The two benchmarks provided by this work – Database-A and Database-B - are finest tool to compare new methods of detection and classifications to the one we show here. There is a room for improvement especially with Database-B. More than that - the question that stands in the heart of this research is classification of larva’s activity to feeding class or non-feeding class. However during feeding process of larvae several other behaviors could be identified. Such behaviors are food spiting and unsuccessful feeding attempt so the question could be extended to classification of larva’s activity to: 1) successful feeding, 2) unsuccessful feeding attempt, 3) spiting, 4) non-feeding activity.

# References

|  |  |
| --- | --- |
| [1] | H. M. Dickinson, T. C. Farley, J. R. Full, M. Koehl, R. Kram and s. Lehman, "How animals mo: an integrative view," *Science,* no. 288, pp. 100-106, 2000. |
| [2] | V. China and R. Holzman, "Hydrodynamic starvation in first-feeding larval fishes," *Proceedings of the national academy of science,* no. 111, pp. 8083-8088, 2014. |
| [3] | R. Holzman, V. China, S. Yaniv and M. Zilka, "Hydrodynamic constraints of suction feeding in low Reynolds numbers, and the critical period of larval fishes," *Integr. Camp. Biol.,* no. 55, pp. 48-61, 2015. |
| [4] | L. P. Hernandes, "Intraspecific scaling of feeding mechanics in an ontogenetic series of zebrafish, Danio rerio.," *J. Exp Biol,* no. 203, pp. 3033-3043. |
| [5] | J. Yamato, J. Ohya and K. Ishii, "Recognizing human action in time-sequential images," in *CVPR*, 1992. |
| [6] | K. Cheung, S. Baker and T. Kanade, "Shape-from-silhouette of articulated objects," in *CVPR*, 2003. |
| [7] | L. Gorelick, M. Blank, E. Shechtman, M. Irani and R. Basri, "Actions as space-time shapes," in *TPAMY 29*, 2007. |
| [8] | S. Sadanand and J. Corso, "Action bank: A high-level representation of activity in Video," in *CVPR*, 2012. |
| [9] | I. Laptev, "On space-time interest points," in *IJCV*, 2005. |
| [10] | S. Lazebnik, C. Schmid and J. Ponce, "Beyond bags of features: Spatial pyramid," in *CVPR*, 2006. |
| [11] | A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time," in *CVPR*, 2010. |
| [12] | J. Liu, Y. Yang, I. Saleemi and M. Shah, "Learning semantic features for action," in *CVIU 116*, 2012. |
| [13] | O. Kliper-Gross, Hassner and W. L. T., "The action similarity labeling challenge.," in *TPAMI 34*, 2012. |
| [14] | S. Ali and M. Shah, "Human action recognition in videos using kinematic features and multiple instance learning," in *TPAMY 32*, 2010. |
| [15] | K. Schindler and L. Gool, "Action snippets: How many frames does human action recognition require?," in *CVPR*, 2008. |
| [16] | Y. Ke, R. Sukthankar and M. Hebert, "Efficient visual event detection using volumetric features," in *ICCV*, 2005. |
| [17] | A. Efros, A. Berg and G. M. J. Mori, "Recognizing action at a distance," in *ICCV*, 2003. |
| [18] | A. Fathi and G. Mori, "Action recognition by learning mid-level motion features," in *CVPR*, 2008. |
| [19] | T. Hassner, Y. Itcher and O. Kliper-Gross, "Violent flows: Real-time detection of violent crowd behavior," in *CVPR*, 2012. |
| [20] | H. Wang, A. Klaser, C. Schmid and C.-L. Liu, "Action Recognition by Dense Trajectories," *CVPR 2011 - IEEE Conference on Computer Vision & Pattern Recognition (2011),* pp. 3169-3176, 2011. |
| [21] | S. N., B. T. and K. K., "Dense point trajectories by GPU-accelerated large displacement optical flows," in *ECCV*, 2010. |
| [22] | V. Kellokumpu, G. Zhao and M. Pietikainen, "Human activity recognition using a dynamic texture based method," in *BMVC*, 2008. |
| [23] | T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classiffcation with local binary patterns," in *TPAMI 24*, 2002. |
| [24] | G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," in *TPAMY 29*, 2007. |
| [25] | L. Yeffet and L. Wolf, "Local trinary patterns for human action recognition," in *ICCV*, 2009. |
| [26] | O. Kliper-Gross, Y. Gurovich, T. Hassner and L. Wolf, "Motion interchange patterns for action recognition in unconstrained videos," *Computer Vision–ECCV 2012,* p. 256–269, 2012. |
| [27] | N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Sys., Man., Cyber. 9 (1): ,* p. 62–66, 1979. |
| [28] | D. G. Lowe., "Distinctive image features from scale-invariant," *IJCV,* vol. 60, no. 2, p. 91–110, 2004. |
| [29] | C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning 20 (3): ,* p. 273, 1995. |
| [30] | D. A. Wolpert, "Stacked generalization," *Neural Networks,* vol. 5, no. 2, pp. 241-260, 1992. |
| [31] | T. Hassner, "A critical review of action recognition benchmarks. In Computer Vision and Pattern Recognition Workshops," *CVPRW,* pp. 245-250, 2013. |

**תקציר**

צילום של פעילות ותנועת בע"ח לגלוי מידע שניתן לכימות הוא כלי שהשימוש בו נפוץ בתחום הביומכניקה. טכנולוגית צילום מתקדמת מאפשרת כעת צילום בקצב גבוה וברזולוציה גבוהה של קטעי ווידאו ארוכים שבהם האירועים המעניינים הם נדירים ולא צפויים. בעוד לאירועים אלו חשיבות אקולוגית רבה, ניתוח של הנתונים בהם האירועים המעניינים הנם נדירים דורש זמן רב, דבר המגביל את הלימוד של השפעתם על כשירות בע"ח וכושרם.

בעזרת השימוש בצילום של פגיות דגים -דגיגים בשלב חיים מוקדם, שלהם מבנה מורפולוגי שונה משמעותית מדג בוגר - התרים אחר מזון, אנחנו מציעים מערכת לזיהוי אוטומטי של תנועת האכילה, פעילות שאינה תדירה אך חיונית לשרידות הדגיגים.

אנחנו משווים את ביצועי הזיהוי של ארבעה מתארים (descriptors) של ווידאו ואת הביצועים של שילובים שונים שלהם לעומת זיהוי ידני של פעולות האכילה. לנתונים שאספנו, מתאר יחיד מציג דיוק של 95-77% בקלאסיפיקציה, ודיוק של 98-88% בזיהוי, תלוי בסוג הדג הנבדק ובגודלו. שילוב של מתארים שונים משפר את דיוק הקלאסיפיקציה ~2%, אבל לא משפר את דיוק הזיהוי.

התוצאות מעידות כי ניתן להקטין משמעותית את המאמץ הנדרש ע"י מומחה לנתח ידנית את קטעי הווידיאו. על המומחה לעבור רק על פעולות האכילה הפוטנציאליות שגילתה המערכת כדי לנקות זיהויים שגויים. בכך השימוש במערכת לזיהוי אוטומטי מפחית משמעותית את מאמץ העבודה הדרוש משבועות של עבודה לשעות בודדות.

דבר זה מאפשר ניתוח של סרטי ווידאו ארוכים ורבים לצורך הרכבת אוסף נתונים גדול שאינו מוטה של פעולות ותנועות רלוונטיות של בע"ח.

**תוכן העניינים**

[**תקציר** 6](#_Toc434095942)

[**1.** **מבוא** 7](#_Toc434095943)

[1.1 רקע 7](#_Toc434095944)

[1.2 בעית זיהוי פעולת האכילה בדגיגים 7](#_Toc434095945)

[1.3 מטרת התזה 8](#_Toc434095946)

[**2.** **עבודות קודמות** 8](#_Toc434095947)

[**3.** **הקמת מערך הניסוי והצילום הדיגיטלי** 10](#_Toc434095948)

[3.1 איפיון הדגיגים 10](#_Toc434095949)

[3.2 הקמת מערך הניסוי 11](#_Toc434095950)

[3.3 זיהוי ידני של פעולות האכילה למטרת התיחסות 12](#_Toc434095951)

[**4.** **זיהוי פעולות האכילה**  12](#_Toc434095952)

[4.1 תיאור כללי 12](#_Toc434095953)

[4.1.1 עיבוד מקדים לווידאו ומציאת מיקום הדגיגים 14](#_Toc434095954)

[4.1.2 סיבוב למצב מנורמל ומציאת מיקום הפה 15](#_Toc434095955)

[4.1.3 יצירת קטעי הווידאו 17](#_Toc434095956)

[4.1.4 ייצוג קטעי הווידאו 17](#_Toc434095957)

[4.1.5 קלאסיפיקציה 21](#_Toc434095958)

[**5.** **תוצאות** 22](#_Toc434095959)

[5.1 בדיקת איכות הקלאסיפיקה 23](#_Toc434095960)

[5.1.1 תוצאות קלאסיפיקציה עבור Benchmark-A 23](#_Toc434095961)

[5.1.2 תוצאות קלאסיפיקציה עבור Benchmark-B 25](#_Toc434095962)

[5.2 בדיקת איכות זיהוי פעולות האכילה 27](#_Toc434095963)

[5.2.1 הליך בדיקת איכות הזיהויים 27](#_Toc434095964)

[5.2.2 תוצאות הזיהוי 27](#_Toc434095965)

[**6.** **סיכום** 30](#_Toc434095966)

[**7.** **רשימת מקורות** 32](#_Toc434095967)

**האוניברסיטה הפתוחה**

**המחלקה למתמטיקה ומדעי המחשב**

**זיהוי תנועת אכילה של פגיות דגים המצולמות במצלמה מהירה**

עבודת תזה זו הוגשה כחלק מהדרישות לקבלת תואר

"מוסמך למדעים" M.Sc. במדעי המחשב

באוניברסיטה הפתוחה

החטיבה למדעי המחשב

ע"י

**אייל שמור**

העבודה הוכנה בהדרכתו של ד"ר טל הסנר

דצמבר 2015

1. Exec file is available at: <http://www.di.ens.fr/~laptev/download.html#stip> [↑](#footnote-ref-1)
2. MATLAB code is available at: [http://www.openu.ac.il/home/hassner/projects/MIP/MIPcode.zip](http://auth.www.openu.ac.il/home/hassner/projects/MIP/MIPcode.zip) [↑](#footnote-ref-2)
3. C-code is available at: <http://lear.inrialpes.fr/people/wang/dense_trajectories> [↑](#footnote-ref-3)
4. MATLAB code is available at: [http://www.openu.ac.il/home/hassner/data/violentflows/](http://auth.www.openu.ac.il/home/hassner/data/violentflows) [↑](#footnote-ref-4)
5. MATLAB Mex file is available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm/ [↑](#footnote-ref-5)