Behavior Recognition

Ariel Iporre

Functional Neuroanatomy

-

Biological Context

Biological study

Characterization of neuronal response to external stimulus from calcium micro-endoscopy images.

Experimental setup 1

The mouse learns to differentiate two textures of a drum (rough and soft) by reinforcement of two different stimulus. Video analysis is focused on whiskering activity.

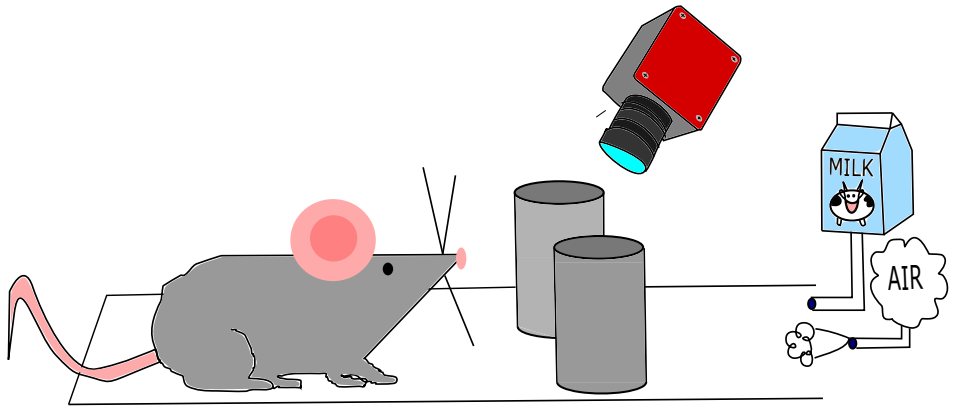


Figure 1: experimental setup

Ariel Iporre

Biological Context

Biological study

Identification of behavioral patterns on open-field basal activity and its evolution

Experimental setup 2

Open field recordings of mice interaction with object and other subjects. Video analysis is focused on behavior clustering.

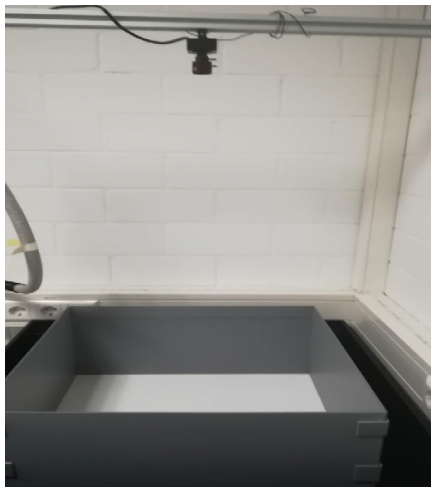


Figure 2: experimental setup

Ariel Iporre

Part 1: whiskering behavior

Problem statement

Aim

Prediction of intended whiskering actions on recordings of mice sensing textures.

Problem

- Identify **whiskering** i.e. coordinated contact and motion of whiskers
- Contact with the drums is some cases may be **unintentional**
- Predict frames containing whisker movement **intented** by the mouse

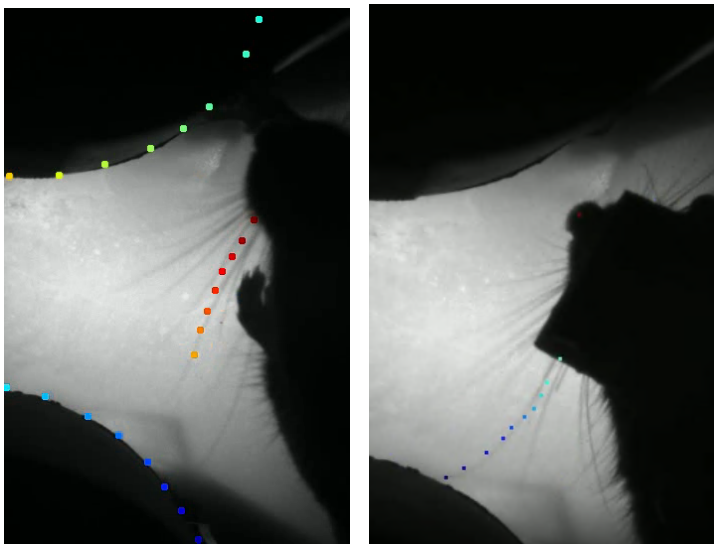
Approach

We employ action recognition to detect two events in the experiment:

1. Whisker contact
2. Whisker motion

Ariel Iporre

Whisker pose estimation



Deep-Lab-Cut is a end-to-end trainable network for pose estimation based on user-defined labels. Two models are trained to:

1. only track whiskers in contact with the drums (used for the whisker contact detection model)
2. continuously track whiskers and nose tip in the video (used for the whisker motion detection model)

What is Deep-Lab-Cut

- **DLC** is presented as an addaptation of the first component of the Deepercut algorithm
- **DeeperCut** is a state-of-art algorithm for multi-person pose estimation made of three elements:
  1. A set of body-part detectors: a variation of deep residual neural with readout layers that predict the location of a body parts
  2. A unary and binary costs calculator: a logistic regression model that computes body part and person assignment probabilities
  3. A pose estimation inference: defined as an Integer Linear Problem (ILP) of labeling assignation of body parts candidates to the people in the image

Deep-Lab-Cut

layer	output size	description
conv1	112x112	7x7,64,stride 2
conv2	56x56	3x3 max-pool, $\begin{bmatrix} 1x1 & 3x3 & 1x1 \\ 64 & 64 & 256 \end{bmatrix} x3$, stride 2
conv3	28x28	$\begin{bmatrix} 1x1 & 3x3 & 1x1 \\ 128 & 128 & 512 \end{bmatrix} x4$, stride 2
conv4	14x14	$\begin{bmatrix} 1x1 & 3x3 & 1x1 \\ 256 & 256 & 1024 \end{bmatrix} x23$, stride 2
conv5	7x7	$\begin{bmatrix} 1x1 & 3x3 & 1x1 \\ 512 & 512 & 2048 \end{bmatrix} x3$, stride 2
output	1x1	average-pool , 1000fc, softmax

Table 1: ResNet-101 architecture details in [He et al. 2015]

- The ResNet-101 has 5 convolutional layers with many convolutional blocks
- Every convolutional block output is concatenated with its input preventing vanishing gradients (aka. skip connections)

Deep-Lab-Cut

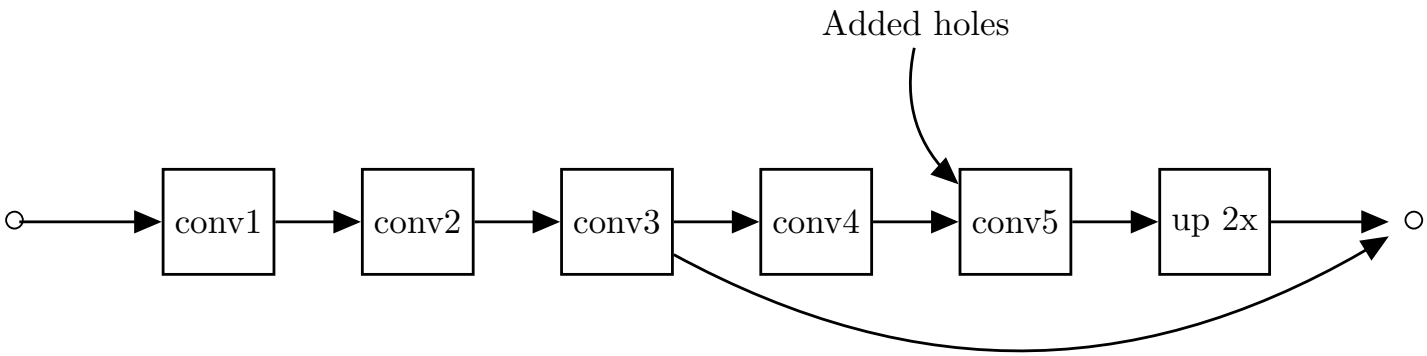


Figure 3: Addaptation of the Resnet to predict body part location maps

The adaptation of the Resnet network involves:

- removing the output layers
- reducing the stride from 2 to 1 and adding holes to the convolution filters of the conv5 layer
- upsampling the output of the conv5 layer with 2x deconvolution and concatenating the output to the output of the conv3 layer

Deep-Lab-Cut

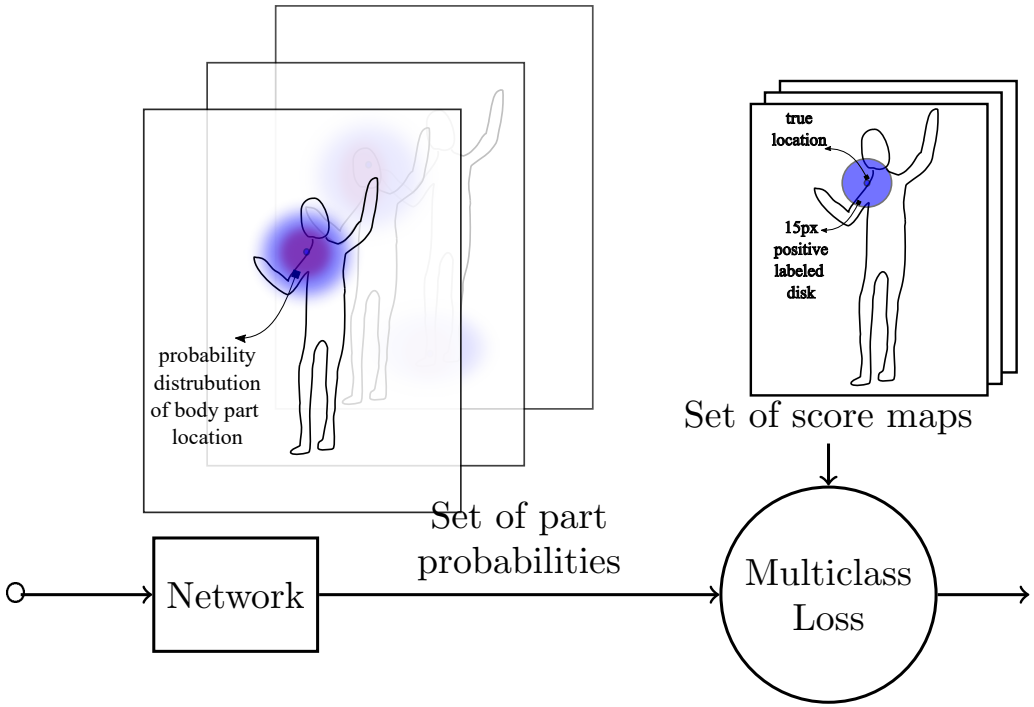
Further modifications to DLC incorporate the other components of the Deepercut algorithm and power up the detection algorithm

Four additional components

1. Score map based classification loss
2. Local refinement
3. Intermediate supervision
4. Pairwise affinity maps

Deep-Lab-Cut

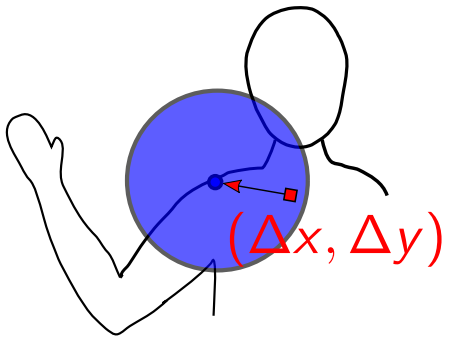
Ariel Iporre



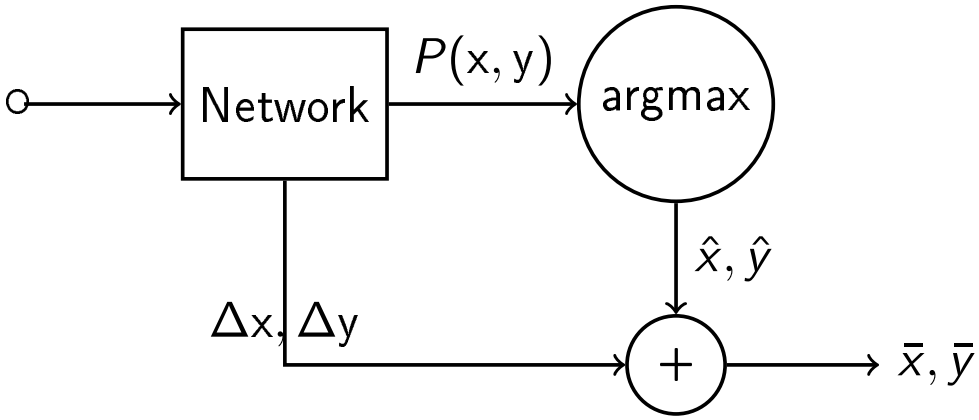
Score map based classification loss

A set of part probabilities at each location, based on a scoremap that assigns positive labels to locations inside a disk of 15px radius to the true location.
The loss associated is a multiclass (classes defined by body parts) logistic loss function

What is Deep-Lab-Cut



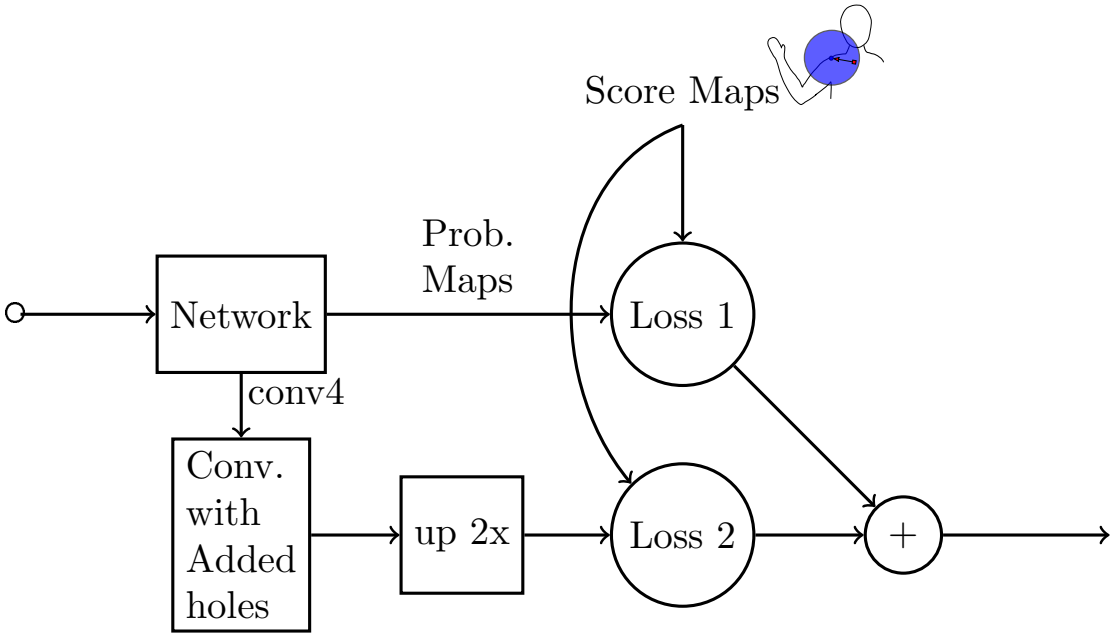
score map offsets



Local refinement

Local refinement is an additional L2 regression (optionally hubber loss) of the offset distances Δx and Δy for all the positive labeled locations on the score-map. The offset is used to refine the estimation of the part location $(\hat{x}, \hat{y}) = argmax_{x,y} P(x, y)$, by adding the estimated offset $(\Delta x, \Delta y)$ at the location $(\hat{x}, \hat{y})$ to get the refined location $(\bar{x}, \bar{y})$.

What is Deep-Lab-Cut



Intermediate supervision

Utilizes the earlier features in network architecture to predict probability-maps. At the output of the conv4 layer adds convolution block with holes and an upsample 2x. The second set of probability-maps are used in a secondary loss helping to encode spatial information between parts. The model is optimized in the sum of such losses.

Ariel Iporre

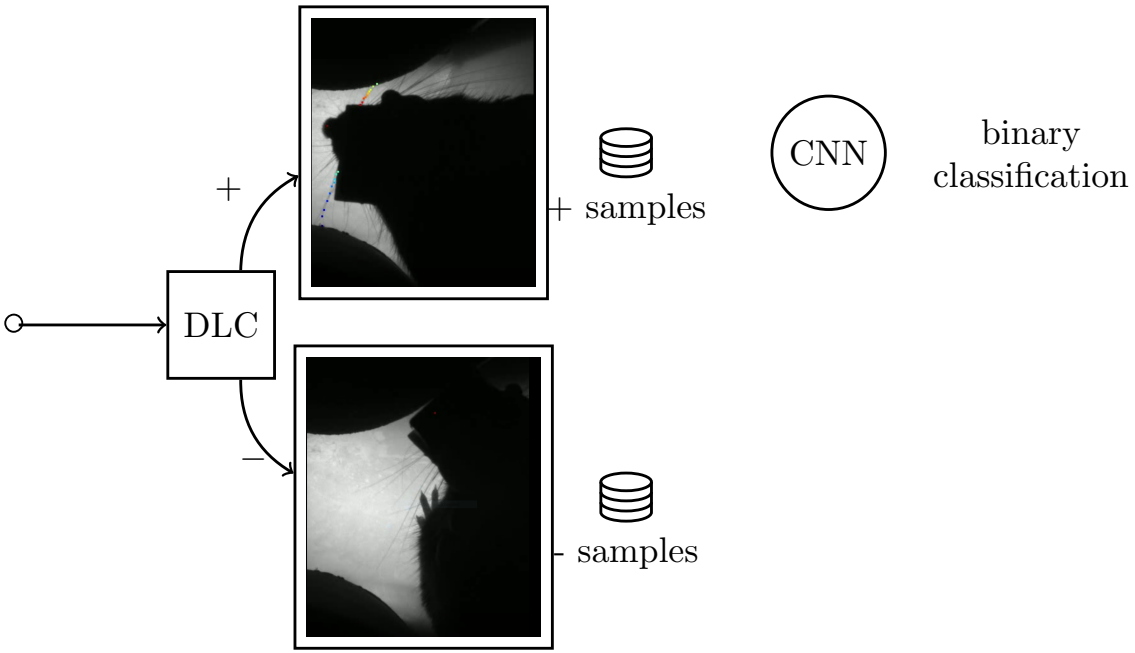
What is Deep-Lab-Cut

Pairwise prediction of affinity maps

Estimation of part affinity maps base of user defined skeletons or $2|C|(|C|-1)$ combinations of the set of body parts C. The part affinity maps are calculated as the offset distances for each location $k = (x_k, y_k)$ at positive labels on the score-map labeled as joint $c \in C$ w.r.t. all $c' \in C/c$ locations.

The model is optimized as a sum of L2 regression losses and multiclass regression losses

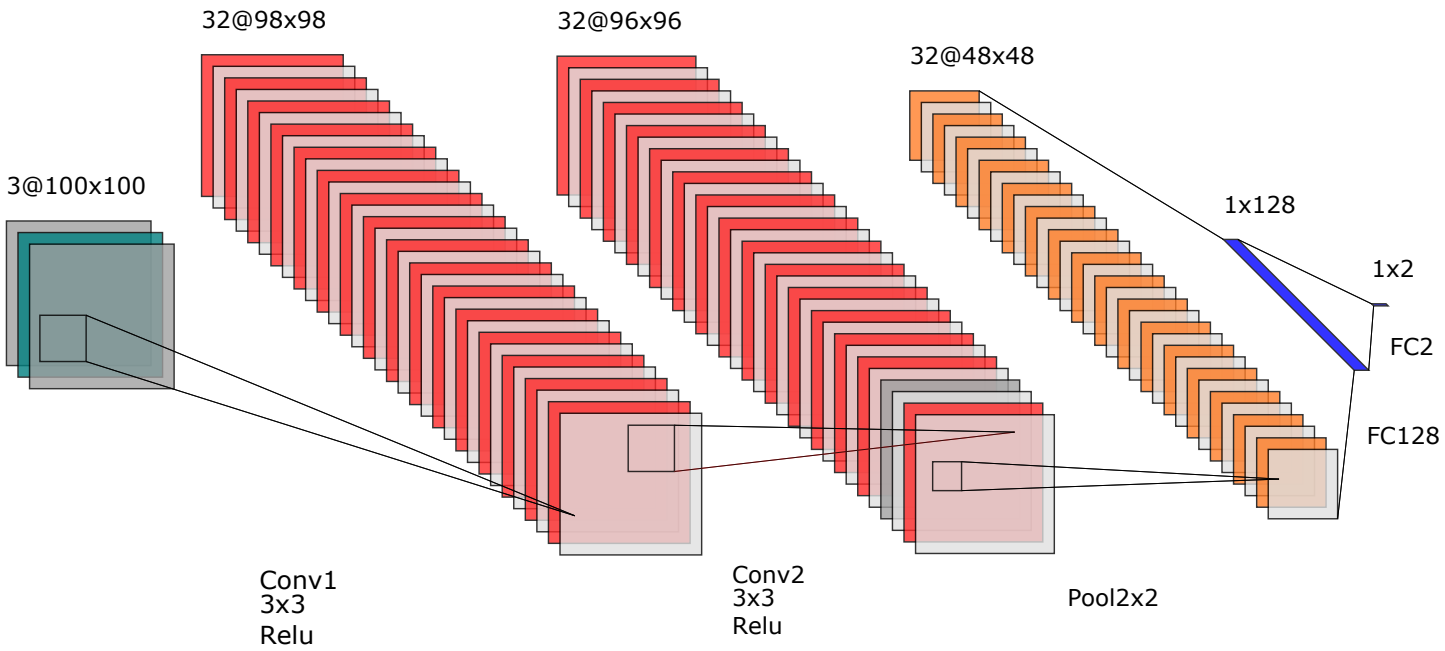
Whisker contact detection



- We trained a secondary CNN to a predict contact probability for every frame in the video
- Training data used for the secondary CNN is produced using a DLC model trained to track only contact whiskers
- Positive samples are frames where at least 10 predicted locations on the whisker have at least 0.8 certainty probability

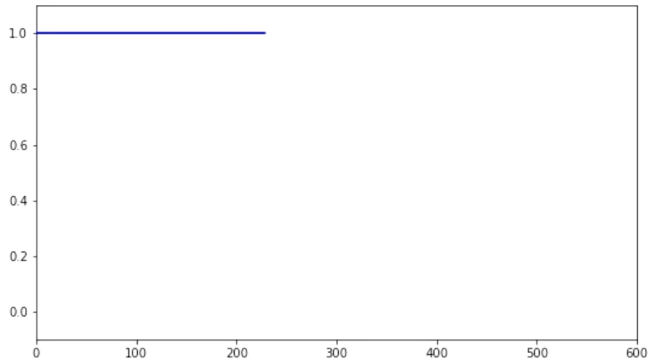
Whisker contact detection

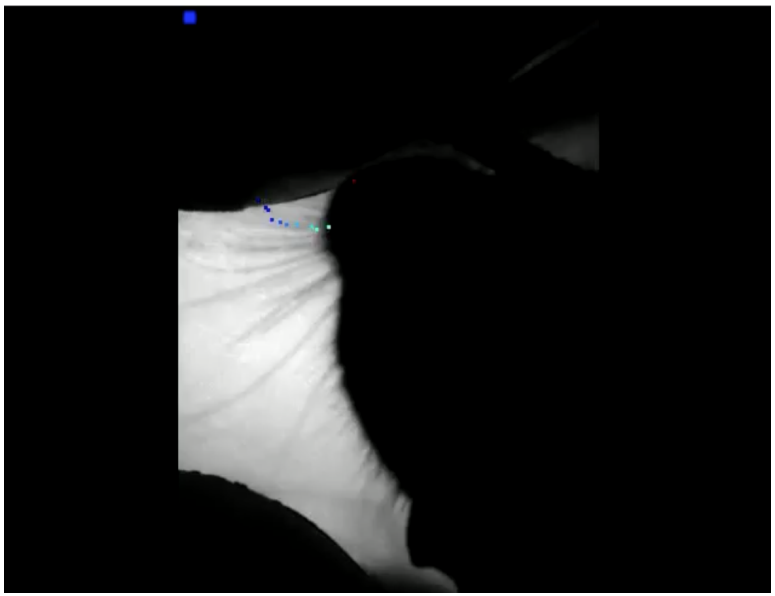
Ariel Iporre



- The secondary CNN learns to remove noisy samples with only one convolutional block and a fully connected layer.

Whisker contact prediction results





- Whisker contact is predicted with a 98% of accuracy.
- Prediction is stable and mostly univocal

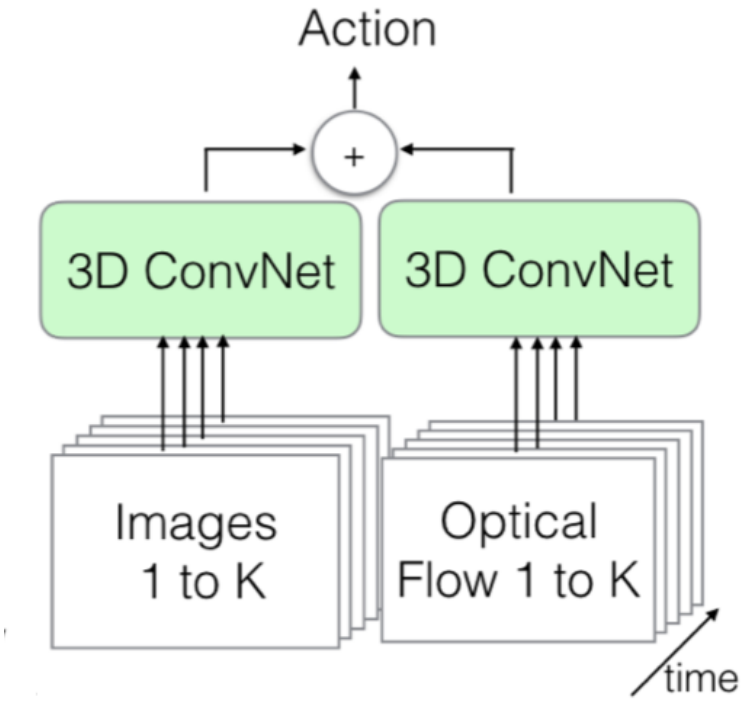
Ariel Iporre

Whisker motion detection

- DLC is used to continuosly detect the whiskers and the snout tip. The predictions are used to analyze the whisker centroid location and calculate its angular velocity w.r.t. snout tip
- From the anglular velocity candidate motion clips of 64, 32 and 10 frames lenght are extracted and manually labeled.
- We implemented an Inception 3D network (I3D) to predict whisker motion

Inception 3D Network

- The I3D combines the motion (optical flow) and apperance (rgb) a video stream using 3D Convolutional Networks.
- The I3D utilizes a 3D propagation of the pretrained imagenet 2D weights as initialization before being trained on the Kinetics dataset.
- Predictions are average on optical flow computed with the TVL1-algorithm and normal RGB frames



Optical flow

$I_0(x)$ $I_1(x+u)$

Optimal flow refers to the perception of motion through the flow of gray pixel values between two images. A displacement vector field u is estimated between two images $I_0(x)$ and $I_1(x+u)$. The optical flow results in a Ill-inverse problem to solve u from the flow continuity equation:

$$\frac{I_1(x+u) - I_0(x)}{\Delta t} = \nabla I \cdot u$$

Optical flow problem

$I_0(x)$ $I_1(x+u)$

The solution used is the duality based on the TVL1 algorithm [6], which solves the disparity map u on the image domain $x \in \Omega$ for the equation:

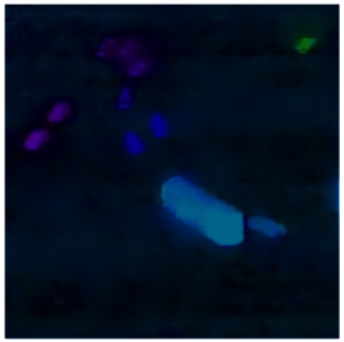
$$min_u \left(\int_\Omega \phi(u) + \int_\Omega \psi(I_1(x+u) - I_0(x)) \right)$$

The first term accounts for the regularization term and the second for the pixel consistency between images. The functions selection is $\phi(\cdot) = \psi(\cdot) = |\cdot|$ as L1-norm.

TVL1 algorithm


$I_0(x)$


$I_1(x+u)$


u

The cost function is approximated with a 1st order Taylor series as

$$min_u \left(\int_\Omega \lambda \underbrace{|I_1(x+u_0) + \Delta I_1 \cdot (u-u_0)|}_{|\rho(u)|} + |\nabla u| \right)$$

and u is solved in a double step optimization:

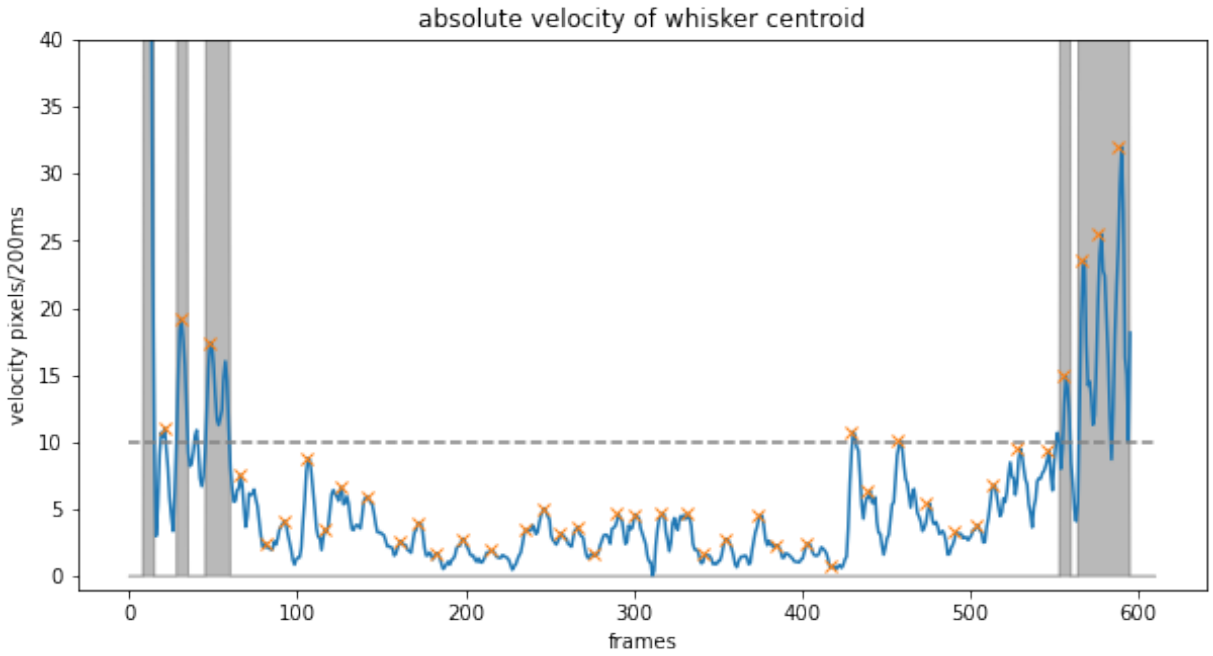
$$min_{u,v} \left(\int_\Omega \lambda |\rho(u)| + |\nabla u| + \frac{1}{2\theta} u v^T \right)$$

Ariel Iporre

Whisker centroid analysis

- To create the dataset, we compute the whisker's centroid and then we find the first derivative to detect motion along the video.
- Frames, where whiskers and nose are not visible, have been excluded from the dataset of positive samples. In gray are regions where the nose is not visible

Whisker motion prediction

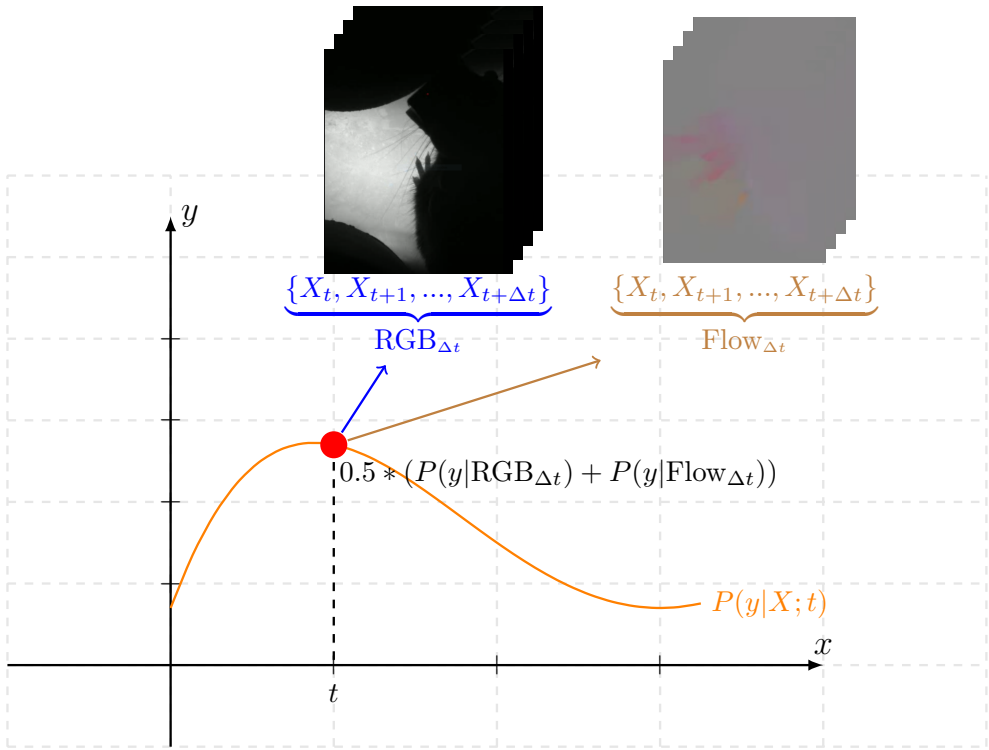


- Based on the absolute velocity of the centroid of the whiskers, clips are randomly sampled from regions where the whiskers are moving
- Additional clips are taken from peaks above 5 pixels/200ms
- Clips are labeled manually with 10 validation iterations where each clip is verified 10 times during different labeling sessions.

Ariel Iporre

Whisker motion prediction

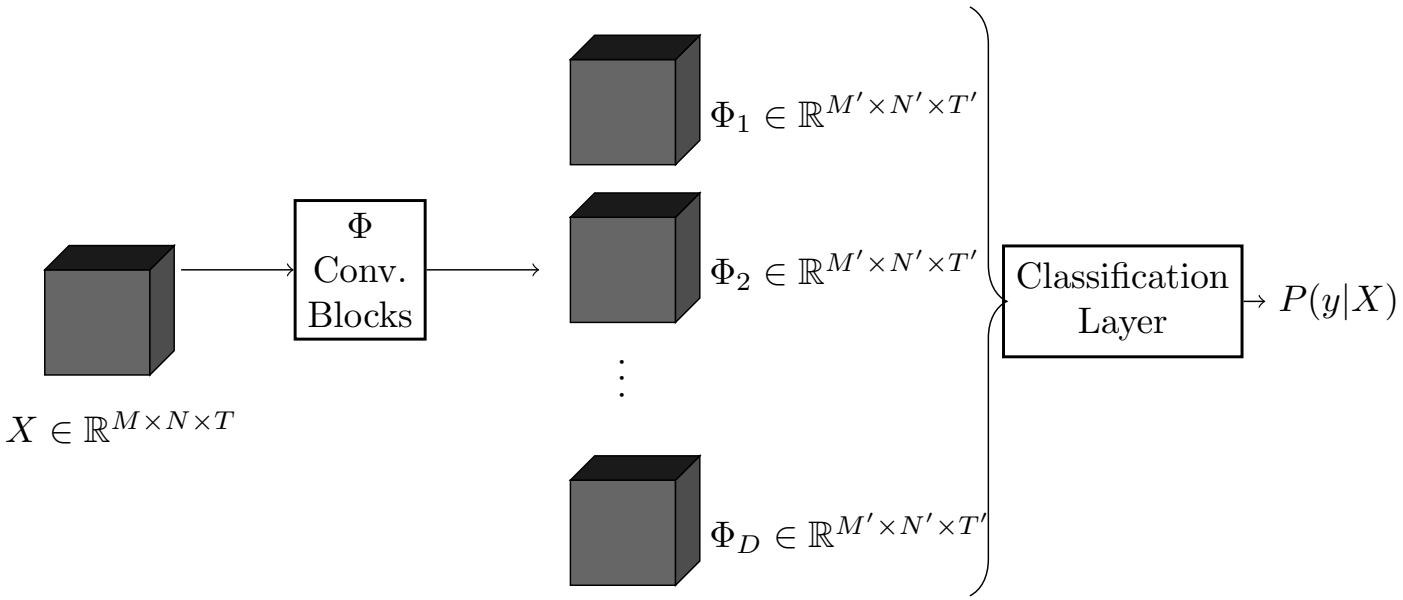
Ariel Iporre



Implementation: Predictions are calculated within a sliding window of 10 frames. Henceforth, the prediction (motion probability signals) considers Δt frames in the future. Finally, the prediction is the average of motion and appearance.

Whisker motion prediction

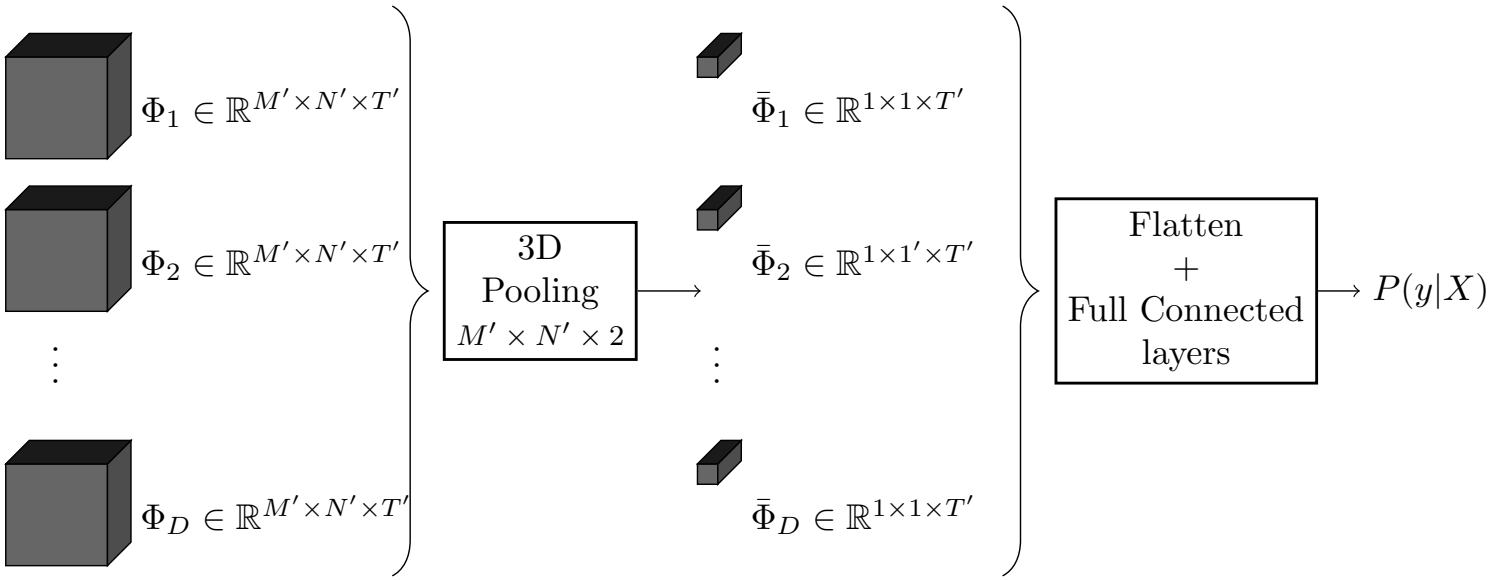
Ariel Iporre



Observation

Based on two implementations of the final classification layer, we observe that the accuracy depends on the aggregation of the spatio-temporal features Φ_i

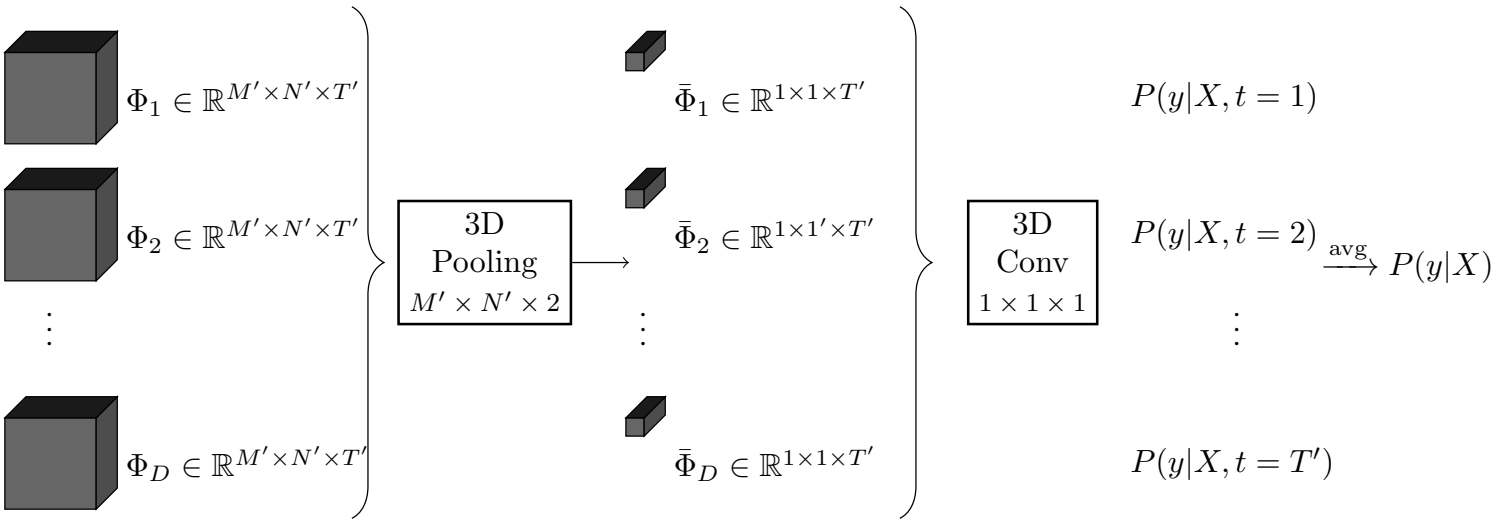
Whisker motion prediction



- The first implementation uses a flatten and a fully-connected layers obtaining a maximum of 81% accuracy
- Features are mixed up since time coordinates are interconnected by the fully-connected layer

Ariel Iporre

Whisker motion prediction



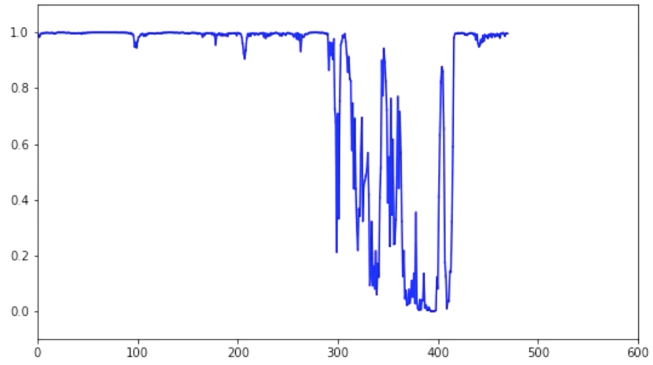
- The second implemenation uses averaged predictions on every remaining time dimension and reaches 91% accuracy
- The 3D Conv. block with a kernel of $1 \times 1 \times 1$ keeps the time coordinates separated and produces a probability estimation for each time.

Ariel Iporre

Whisker motion prediction results

Ariel Iporre

- We observe a time shift effect of $+10$ frames in the probability signal $P(y = \text{motion}|X; t)$ with respect to the event time due to implementation
- We obtained an accuracy of 91% combining motion and appearance video streams



Part 2: Behavior clustering

Behavior clustering

- Find clusters of behavior given unlabeled data. Unsupervised learning actions in behavior videos.

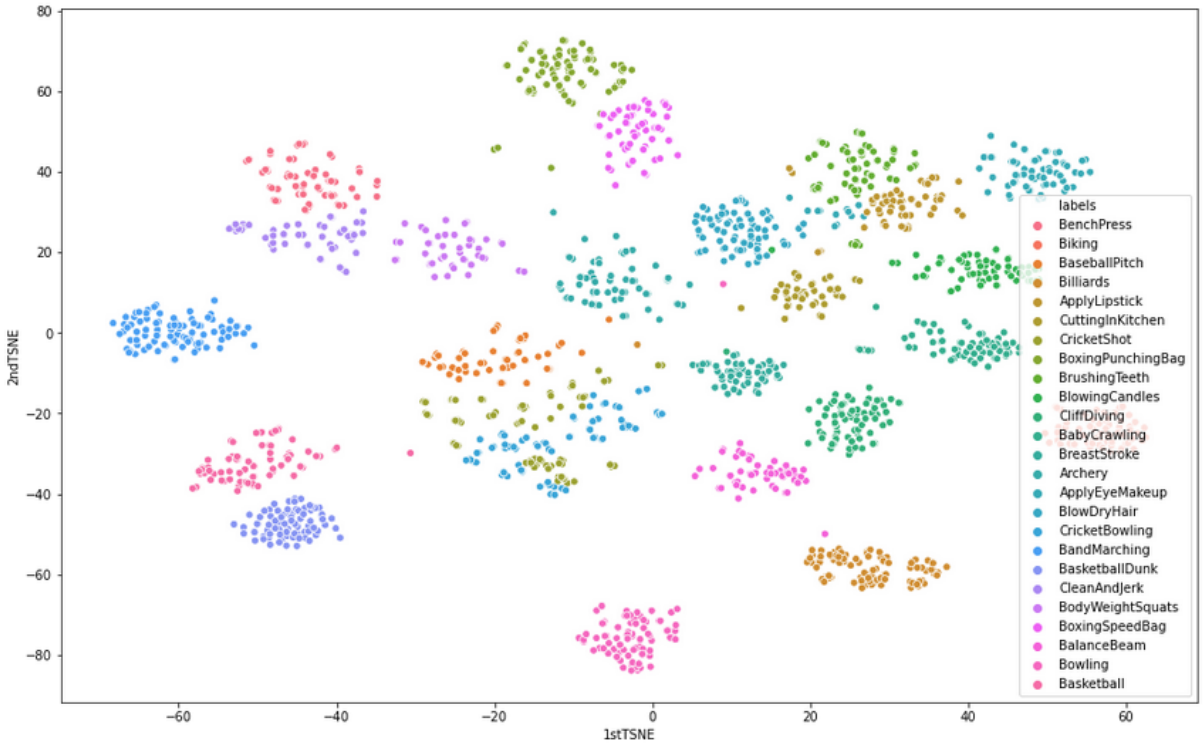
Questions

- How to obtain meaningful features from behavior videos?
- How to aggregate features into a fixed length feature representation? (considering the problems identified in the whiskering prediction task above)
- How to obtain a space of representation of behavior videos where it is possible to form meaningful clusters?

How to obtain meaningful features?

- We need features that encode the action into a separable space.
- Successful feature extraction captures **motion and appearance**, as shown to be necessary components on action classification and video understanding [1, 5]

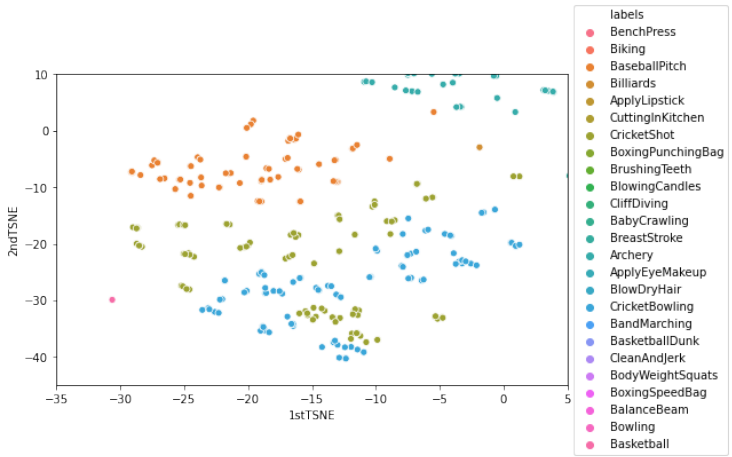
How to obtain meaningful features?



- For videos in a subset of the UCF101 dataset, we visualize the 3D pooling output tensors $[\bar{\Phi}_1, \bar{\Phi}_2, ..., \bar{\Phi}_D]$ from the I3D network embedded in 2 dimensions with the TSNE algorithm
- Clusters are distinctive for many of the actions (25 classes)

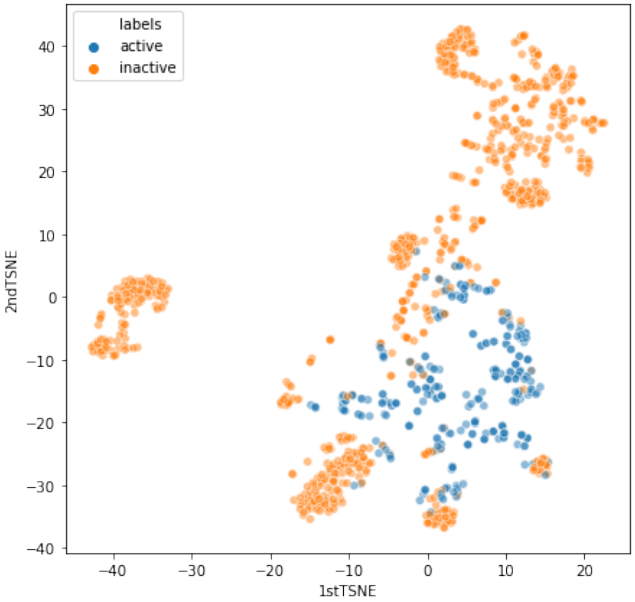
Ariel Iporre

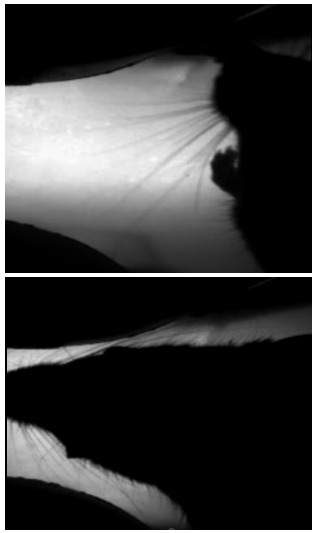
How to obtain meaningful features?



- Some classes are more or less mixed for example BaseballPitch, CricketShot and CricketBowling.
- It implies that feature extraction of the I3D convolution block (spatio-temporal features) are focused on appearance

How to obtain meaningful features?





active

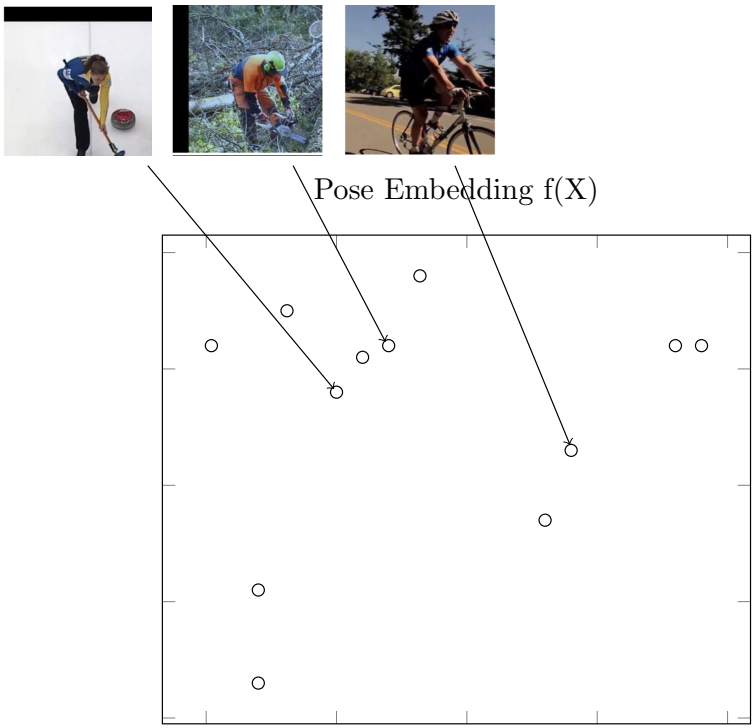
inactive

Observartion

- We observe the same effect on the whisker motion dataset
- If appearance is a too dominant feature, then generalization will require many labeled samples (supervised learning), which makes it difficult to extend to other domains. **Pose embedding may be a better approach to encode actions**

Ariel Iporre

How to obtain meaningful features?

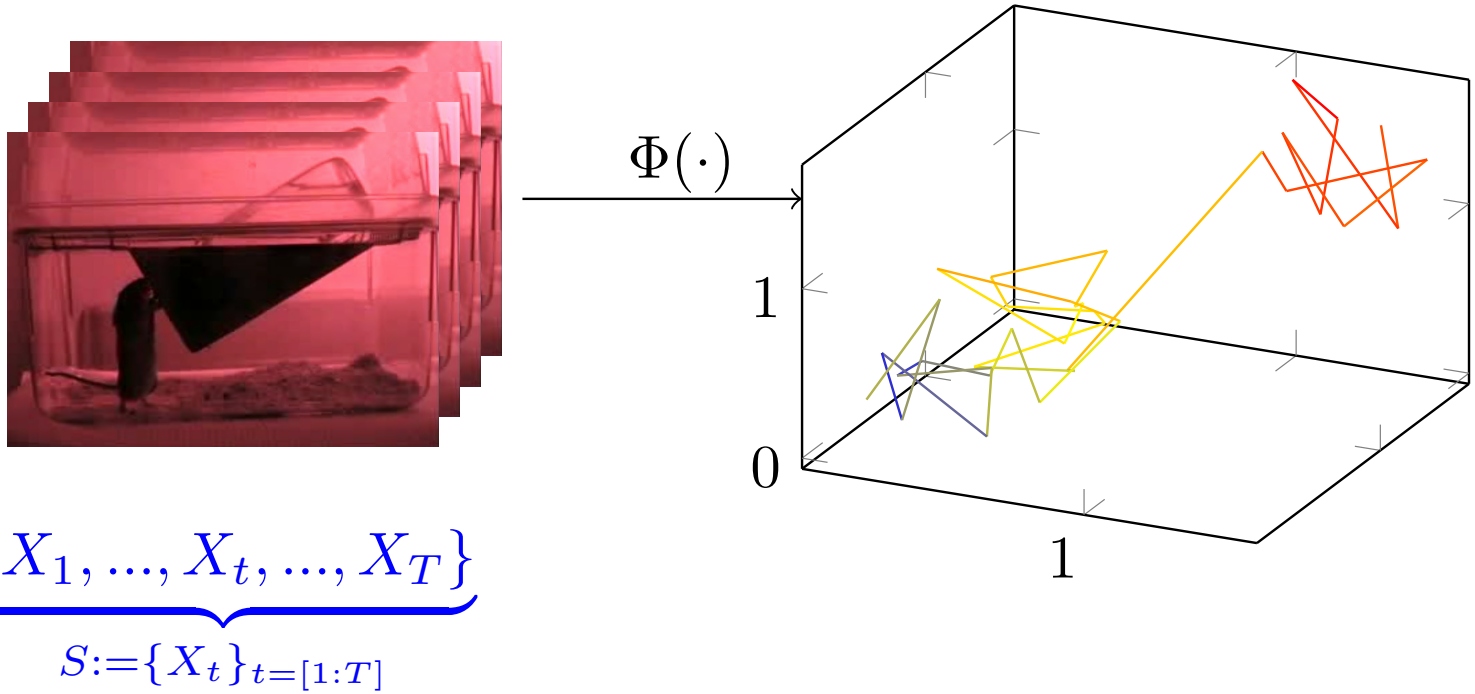


- Find a pose embedding space that places images of similar pose nearby. That is, if two images X_i and X_j, we want to learn a pose function $\Phi : X \rightarrow \Phi(X)$ s.t. distance of similar pose embeddings are minimized while different poses are maximized.

Ariel Iporre

How to obtain meaningful features?

Ariel Iporre

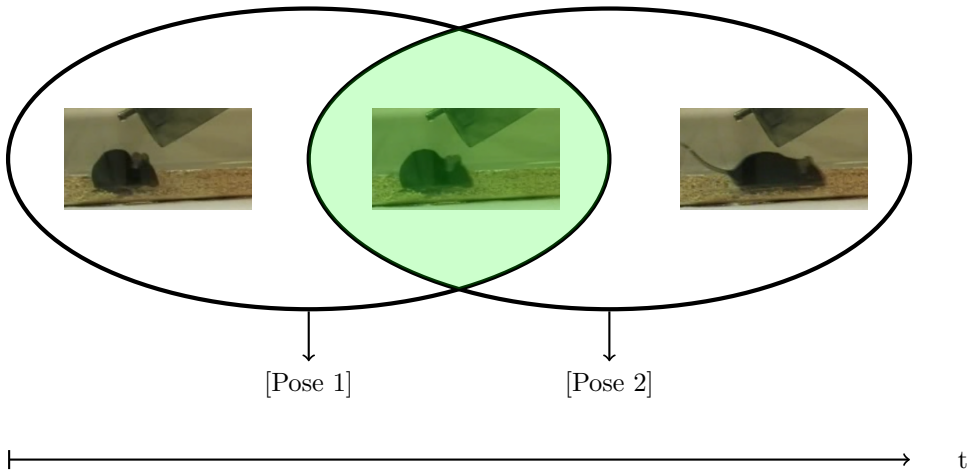


- The pose function $\Phi : X \rightarrow \Phi(X)$ will map a set sequence images $S = \{X_t\}_{t=[1:T]}$ to a set of pose transitions $\hat{S} = \{\Phi_t\}_{t\in[1:T]}$ that interconnected form a smooth D-dimensional curve.

Pose embedding

- As proposed in [3], we aim to obtain a pose embedding space that learns:
  - Structured representation of posture
  - Temporal development
- This is done in [3] by mapping two sequences $S = \{X_j\}$ and $S' = \{X_{j'}\}$ with $\pi : S \rightarrow S'$ that matches similar apperance s.t. frames are chronically ordered, one-to-one and invariant to frame rate
- How to learn the function π alternately to use ILP?

Hypothesis pose transition



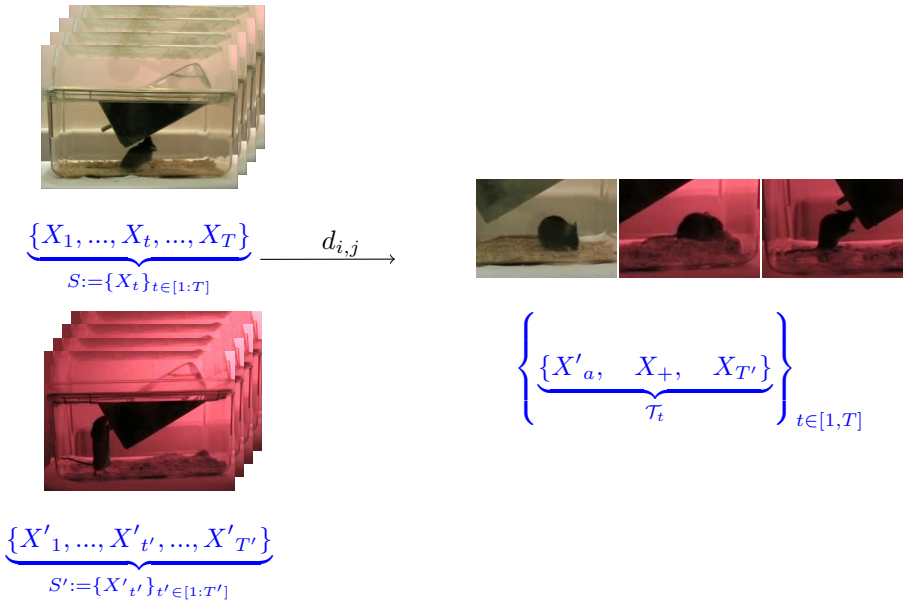
Hypothesis

Action is composed by finite pose milestones interconnected by finite pose transitions distributed on time such that:

- Pose clusters are distributed over time
- Similar poses have similar pose embeddings and are close in time i.e. $\Phi_t \approx \Phi_{t'} \rightarrow t \approx t'$. Conversely, different poses are separated on time i.e. $|\Phi_t - \Phi_{t'}| >> 0 \rightarrow |t - t'| >> 0$

Ariel Iporre

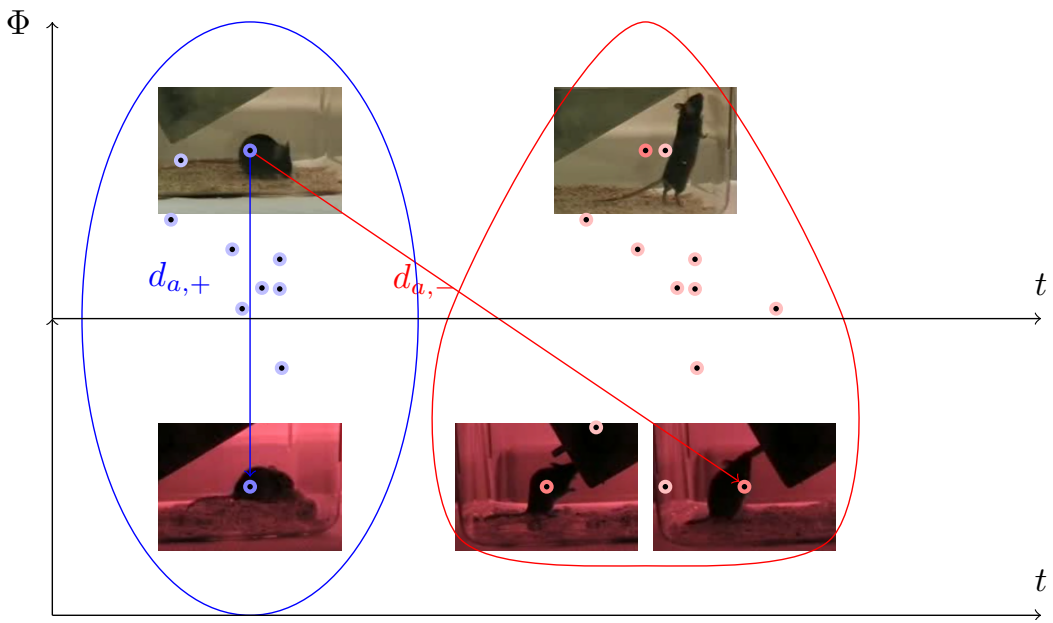
Pose transition function



- We employ alternately a unsupervised cluster index function $\pi_k : \{S \cap S'\} \rightarrow \{\tau_k\}_{k \in [1,K]}$ with the metric $d_{i,j} = ||\Phi_i(x) - \Phi_j(x)||_2^2 + |i - j|$
- As π in [3], π_k is used to form triplets $\mathcal{T} = \{X_a, X_+, X_-\}$, s.t. $\Phi_a \approx \Phi_+$ and $\Phi_a \not\approx \Phi_-$
- Φ is optimized to with the loss function $L(\{\mathcal{T}_t\}) := \frac{1}{T} \sum_t L'(\mathcal{T}_t) \quad L' := [||\Phi_a - \Phi_+||_2 - ||\Phi_a - \Phi_-||_2 + \delta]_+$ and δ hyper-parameter

Ariel Iporre

Pose transition function



- X_+ is select from the sequence S' where $d_{a,+}$ is minimum in the same cluster.
- X_- is obtained by sorting and selecting the n-th furthest frame X to the archor frame in different clusters. Further frames are easier to learn.
- Curriculum learning closes gab between pose embeddings

Algorithm unsupervised pose learning

Algorithm 1: Training pose embedding network

Init: Set weights θ of CNN layers Φ ;

Result: Optimized pose embedding map $\Phi(\cdot)$

while *Loss* $L(\{\mathcal{T}_t\})$>*threshold* **do**

1: Sample two video clips S and S' from the dataset;

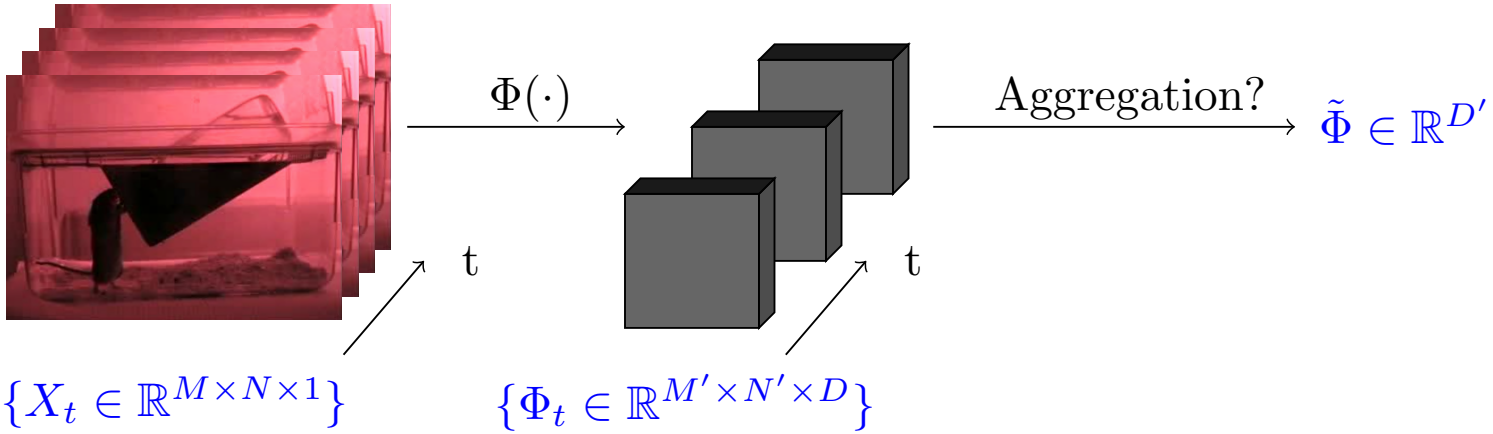
2: Compute the cluster index function π_k w.r.t. pose distance function $d_{i,j}(\Phi_i(X), \Phi_j(X))$ (Unsupervised Learn Pose Correlations);

3: Compute a sample set of triplets $\{\mathcal{T}_t\}_{t\in[1,T]}$;

4: Optimize weights θ of pose embedding map $\Phi(\cdot;\theta)$ w.r.t. triplet's loss $L(\{\mathcal{T}_t\})$;

end

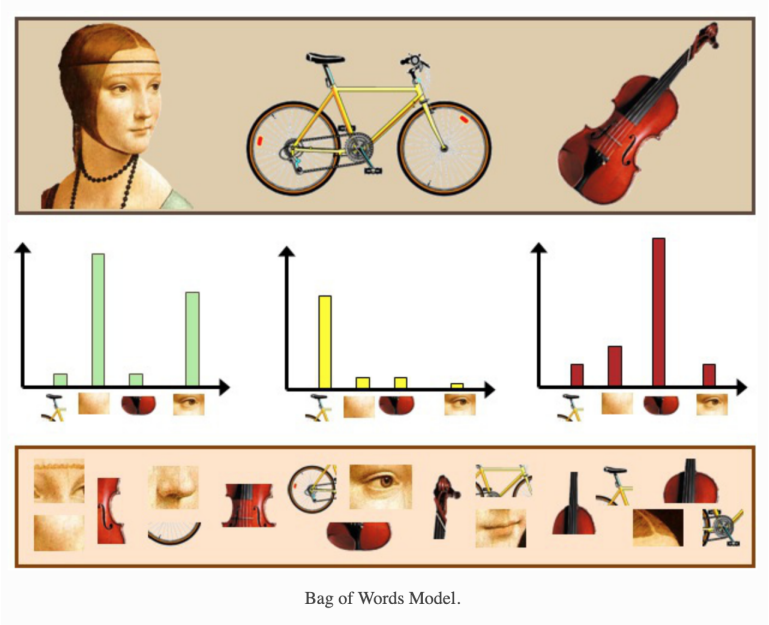
How to aggregate fixed length feature representation?



- Features are obtained from behavior videos $\{X_t\}$ using pose embedding $\Phi(\cdot)$, but what we need is a **global feature descriptor such features over time are integrated**
- The VLAD algorithm has been used to create a fixed length descriptors from dense trajectories (e.g. HOG transform).
- Action VLAD aggregation layers uses the idea of VLAD-algorithm [2]

VLAD

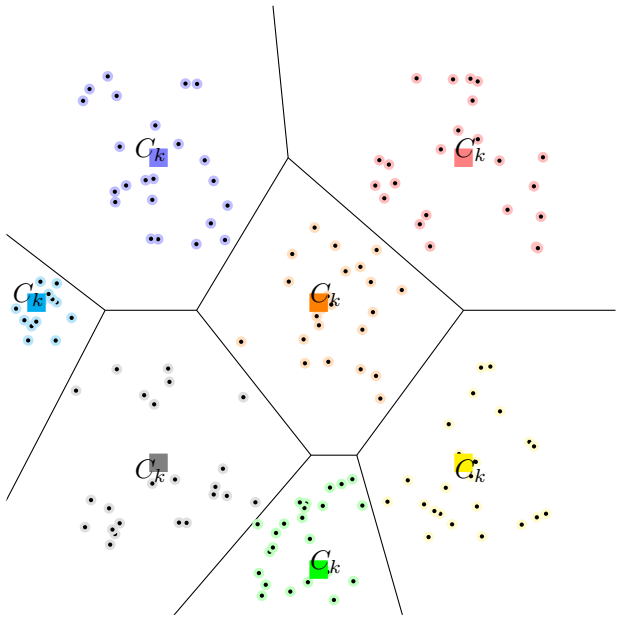
Vector Locally Aggregated Descriptors (VLAD) is an extension of a visual bag-of-words algorithm:



Bag of Words Model.

1. Compute features from patches of the input image
2. Cluster features
3. Compute histograms as a fixed length descriptor

VLAD



Algorithm 2: VLAD algorithm

Init: Set of feature $d-$dimensional vectors x;

Result: Set of v^k of D-dimensional vectors for each cluster C_k

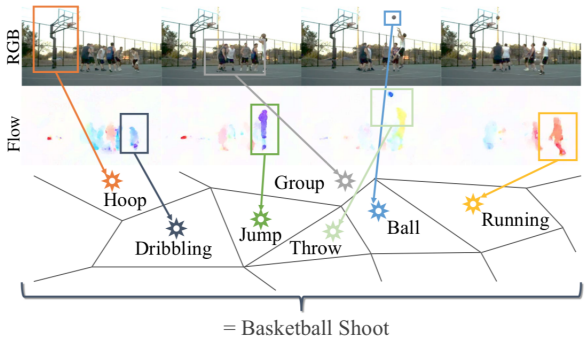
1: Assess discrepancy of samples within clusters C_k as Nearest Neighbor $x \in NN(x)$;

2: Each cluster is aggregated as:

$$v^k = \sum_{x \in NN(x)} (x - c_k)$$

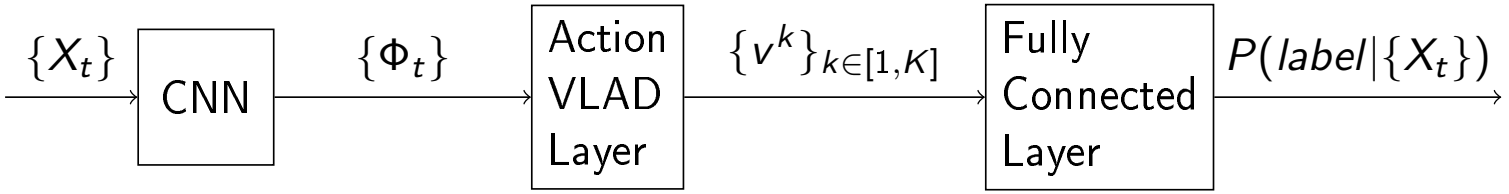
Ariel Iporre

Action VLAD



- Description of actions requires an appropiated spatio-temporal representation that captures evidence over the video.
- For example, basketball shot is described by different events (Hoop, Jump, Dribbling, etc.) happening at different time points.
- Action VLAD proposes a polling method to represent features along space and time

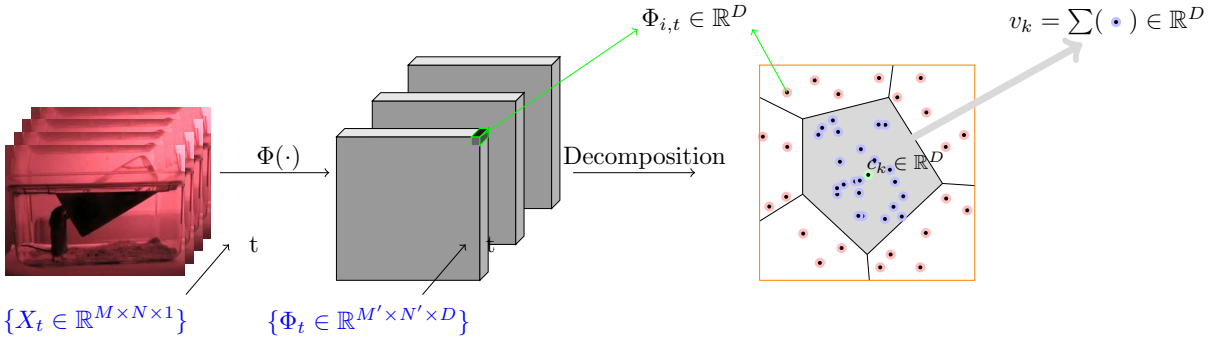
Action VLAD



How does it work?

1. Sample T frames from a RGB (appearance) and flow (motion) streams forming a the input sequence $S = \{X_t\}_{t\in[1,T]}$.
2. Features are computed for each frame i.e. build a spatio-temporal feature extraction $\Phi : \{X_t\}_{t\in[1,T]} \rightarrow \{\Phi_t\}_{t\in[1,T]}$ with $\Phi_t \in \mathbb{R}^{M\times N\times D}$
3. The Action VLAD layer makes the aggregation of extracted features into K descriptors: v^k with $k \in [1, K]$
4. The concatenated descriptors are input to a Fully Connected layer to estimate the *a posteori* probabilities for labeled actions $P(label|\{X_t\})$

Action VLAD



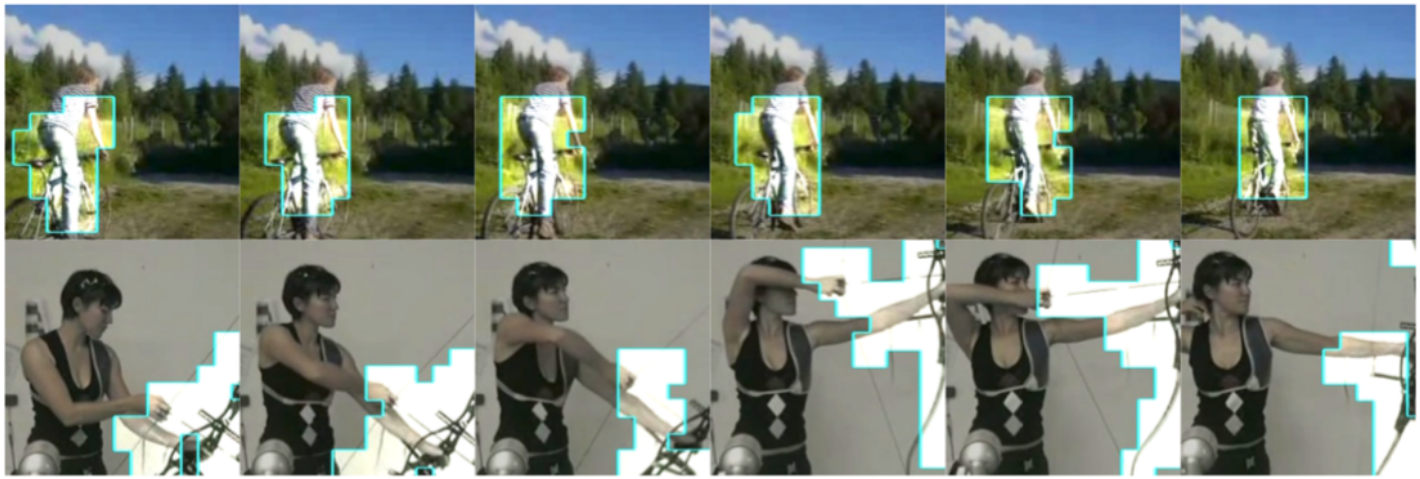
- The set of feature vectors denoted $\mathcal{R}^D := \{\Phi_{i,t}\}_{i \in [1, M' * N'], t \in [1, T]}$ is formed from the affine decomposition of $\{\Phi_t\}$
- A set of K cluster cells is defined $\mathcal{C}_k$, s.t. $\mathcal{R}^D = \bigcup_k \mathcal{C}_k$
- The aggregated descriptors v^k are D-dimensional vectors then

$$v^k = \sum_{i,t} \tilde{a}_k(x_{i,t})(\Phi_{i,t} - c_k)$$

- $\tilde{a}_k$ is soft assignment function for the sample $\Phi_{i,t}$ within the cluster k and c_k is the pseudo-centroid of the cluster

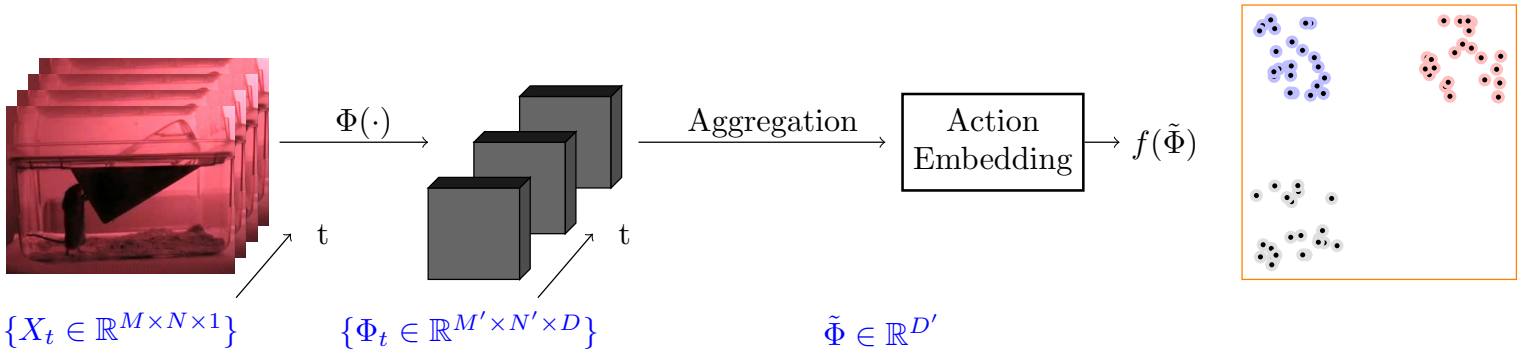
Ariel Iporre

Action VLAD



- The VLAD vectors tracks the patches of movement and captures the motion of an object along the video
- At the end, each *video sequence* is then encoded as K*D-dimensional vector properly representing events, which is an advantage as the spatio-temporal features $\Phi_{i,t}$ cannot be just concatenated to form such vector.

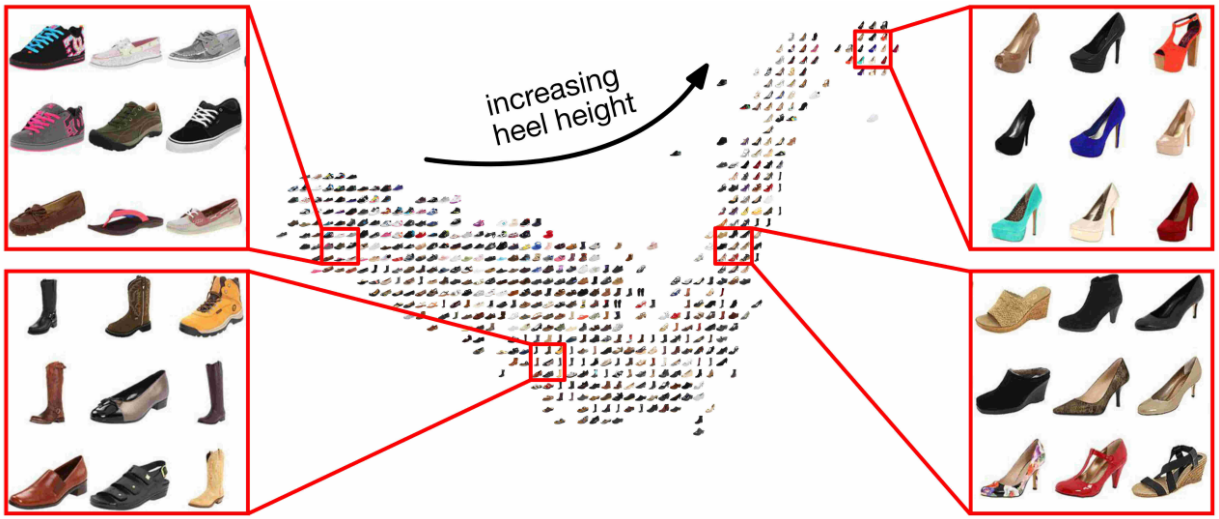
How to obtain a space of representation of behavior videos where it is possible to form meaningful clusters?



- We represent the set of pose embeddings $\{\Phi_t\}$ as a single action embedding $f(\tilde{\Phi}) \in \mathbb{R}^{D''}$
- For the implementation, the experiment video is divided into clips of different lengths forming a behavior video dataset. Then action embedding representations are computed for each behavior video.
- The action embedding map f must be well defined, making it easy to identify clusters. It can be achieved with **metric learning**

Ariel Iporre

Metric learning

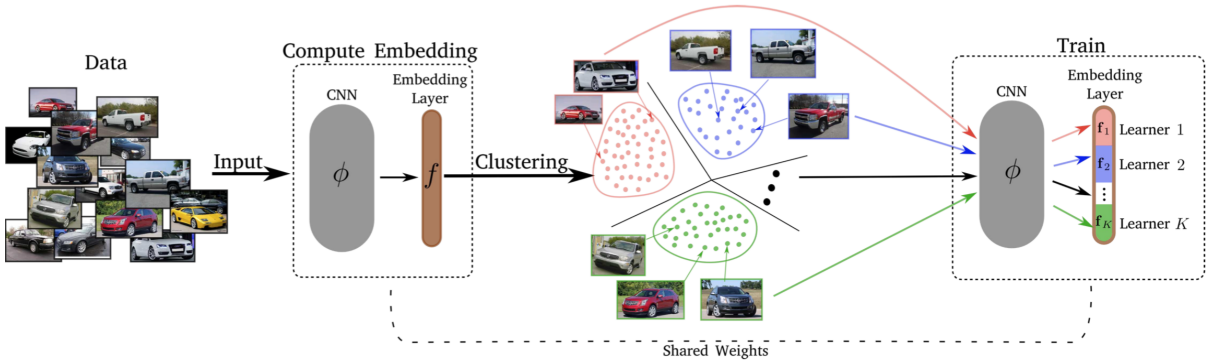


- Metric learning finds an embedding space which separates dissimilar samples and gathers similar samples
- The task employs a distance metric between pairs of samples of behavior videos expressed as aggregated feature representations $\tilde{\Phi}_i$ and $\tilde{\Phi}_j$ such that

$$d_f(\tilde{\Phi}_i, \tilde{\Phi}_j) = ||f(\tilde{\Phi}_i) - f(\tilde{\Phi}_j)||_2^2$$

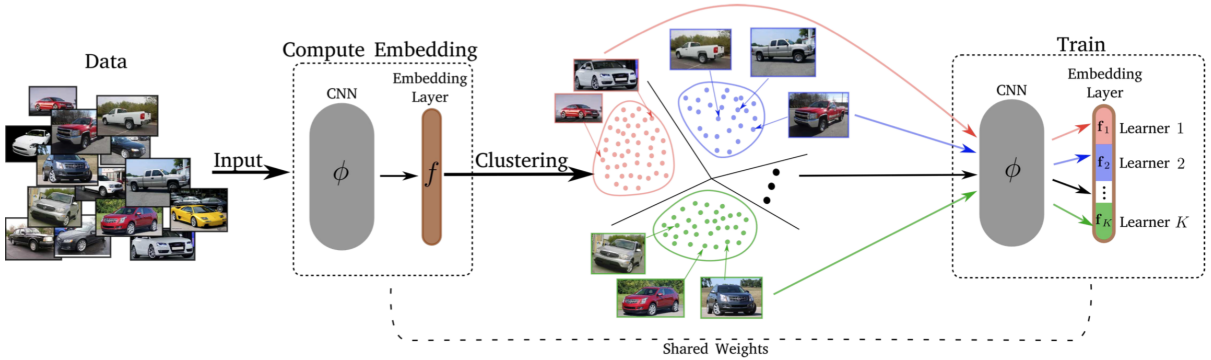
Ariel Iporre

Divide and conquer: semi-supervised learning approach for metric learning



- Divide and conquer is a semi supervised approach that subdivides the space in K-subsets and learns an embedding sub-space for each cluster (**divide**). The original embedding space is constructed by putting together all learners as a concatenated vector (**conquer**) [4].

Divide and conquer: semi-supervised learning approach for metric learning



- Given a set of images, the embedding space is the composition of a CNN ϕ and a embedding layer f. The objective is to learn $\Phi \circ f$ with a metric $d_{\Phi \circ f}(\cdot, \cdot)$
- In the embedding space, the images are first clustered into K-clusters$\{C_k\}_{k \in [1,K]}$
- Each cluster is associated with a learner that models D/K dimensions from the embedding space
- Learners are trained alternating one at the time sharing the weights associated with the CNN ϕ

Ariel Iporre

Behavior clustering algorithm

The proposed behavior clustering algorithm incorporates the solutions:

Methods

- **Features optimized for action representation** are obtained by computing pose embedding for each frame of clips of random length from the experiment video
- The **fixed length feature representation** problem is addresed by incorporating a VLAD aggregation layer
- **An optimized space of representation of behavior videos** is solved with the divide-and-conquer approach for metric learning

Datasets

We consider following datasets:

Datasets

- **CRIM13 Caltech Resident-Intruder Mouse dataset**, which has 237 videos of 10 minutes recorded from top- and side- views using two fixed, synchronized cameras of one or two mice in a cage
- **MIT CSAIL's dataset for Automated Mouse Behavior Recognition**, 4200 clip of 8 stereotypical behaviors (eat, drink, groom, hang, micromovement, rear, rest, and walk)

References

Joao Carreira and Andrew Zisserman.

Quo vadis, action recognition? A new model and the kinetics dataset.

In *proceedings of the IEEE CCVPR*, 2017.

Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell.

ActionVlad: Learning spatio-temporal aggregation for action classification.

In *Proceedings of the IEEE CCVPR*, 2017.

Timo Milbich, Miguel Bautista, Ekaterina Sutter, and Bjorn Ommer.

Unsupervised video understanding by reconciliation of posture similarities.

In *Proceedings of the IEEE ICCV*, 2017.

Artsiom Sanakoyeu, Vadim Tschernezki, Uta Büchler, and Björn Ommer.

Divide and conquer the embedding space for metric learning.

In *Proceedings of the IEEE CCVPR*, 2019.

Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville.

Describing videos by exploiting temporal structure.

In *Proceedings of the IEEE ICCV*, 2015.

Christopher Zach, Thomas Pock, and Horst Bischof.

A duality based approach for realtime TV-L1 optical flow.

volume 4713, pages 214–223, 09 2007.