A Neuro-Symbolic Approach to Structured Event Recognition

Gianluca Apriceno [1,2], Andrea Passerini[2], Luciano Serafini[1]

[1]Fondazione Bruno Kessler, Trento, Italy
[2]University of Trento, Trento, Italy

28th International Symposium on Temporal Representation and Reasoning

April 8, 2022

FONDAZIONE BRUNO KESSLER

UNIVERSITY OF TRENTO

Table of contents

AI war

Symbolic vs Subsymbolic AI

Symbolic:

- **Pros**:
- **Cons**:

Subsymbolic:

- **Pros**:
- **Cons**:

Symbolic vs Subsymbolic AI

Symbolic:

- 50s - 80s
- based on high-level symbols
- Examples:
  - Logic programming
  - Semantic nets
  - Production rules
- **Pros**:
- **Cons**:

Subsymbolic:

- **Pros**:
- **Cons**:

Symbolic vs Subsymbolic AI

Symbolic:

- 50s - 80s
- based on high-level symbols
- Examples:
  - Logic programming
  - Semantic nets
  - Production rules
- **Pros**:
  - reasoning + derivation of new knowledge, interpretable, ...
- **Cons**:

Subsymbolic:

- **Pros**:
- **Cons**:

Symbolic vs Subsymbolic AI

Symbolic:

- 50s - 80s
- based on high-level symbols
- Examples:
  - Logic programming
  - Semantic nets
  - Production rules
- **Pros**:
  - reasoning + derivation of new knowledge, interpretable, . . .
- **Cons**:
  - expert knowledge, no robust to noise, scalability issues, . . .

Subsymbolic:

- **Pros**:
- **Cons**:

Symbolic vs Subsymbolic AI

Symbolic:

- 50s - 80s
- based on high-level symbols
- Examples:
  - Logic programming
  - Semantic nets
  - Production rules
- **Pros**:
  - reasoning + derivation of new knowledge, interpretable, ...
- **Cons**:
  - expert knowledge, no robust to noise, scalability issues, ...

Subsymbolic:

- 80s - up to now
- implicit representation
- Examples:
  - Bayesian learning
  - Neural Network
  - Deep learning
- **Pros**:
- **Cons**:

Symbolic vs Subsymbolic AI

Symbolic:

- 50s - 80s
- based on high-level symbols
- Examples:
  - Logic programming
  - Semantic nets
  - Production rules
- **Pros**:
  - reasoning + derivation of new knowledge, interpretable, ...
- **Cons**:
  - expert knowledge, no robust to noise, scalability issues, ...

Subsymbolic:

- 80s - up to now
- implicit representation
- Examples:
  - Bayesian learning
  - Neural Network
  - Deep learning
- **Pros**:
  - learning from data, robust to noise, scalable, ...
- **Cons**:

Symbolic vs Subsymbolic AI

Symbolic:

- 50s - 80s
- based on high-level symbols
- Examples:
  - Logic programming
  - Semantic nets
  - Production rules
- **Pros**:
  - reasoning + derivation of new knowledge, interpretable, ...
- **Cons**:
  - expert knowledge, no robust to noise, scalability issues, ...

Subsymbolic:

- 80s - up to now
- implicit representation
- Examples:
  - Bayesian learning
  - Neural Network
  - Deep learning
- **Pros**:
  - learning from data, robust to noise, scalable, ...
- **Cons**:
  - Black box, data hungry, no reasoning capability, ...

Neuro-Symbolic

Neuro-Symbolic

- Integration of symbolic and subsymbolic:

- Examples:

Neuro-Symbolic

- Integration of symbolic and subsymbolic:
  - Less training data
  - Low level processing with high level reasoning
  - Constraint predictions
  - Interpretable
- Examples:

Neuro-Symbolic

- Integration of symbolic and subsymbolic:
  - Less training data
  - Low level processing with high level reasoning
  - Constraint predictions
  - Interpretable
- Examples:
  - **Fuzzy logic**: Logic Tensor Network and LYRICS
  - **Probabilistic graphical models**: Deep Structured Models and Deep Logical Models
  - **LP + Probabilistic reasoning**: NeurASP and **DeepProbLog**

Prolog - ProbLog - DeepProbLog

Logic Programming:

- **Facts**:

  unconditionally true statements on both object and their relations

- **Rules**:

  relations among objects

- **Queries**:

  interaction with the logic program

Logic Programming:

- **Facts**:
  unconditionally true statements on both object and their relations
- **Rules**:
  relations among objects
- **Queries**:
  interaction with the logic program

Prolog

```
#facts
  head1. head2.

#rule
 twoHeads:-
   head1,
   head2.

#query
 query(twoHeads).

True
```

Logic Programming:

- **Facts**:

  unconditionally true statements on both object and their relations

- **Rules**:

  relations among objects

- **Queries**:

  interaction with the logic program

ProbLog

```
#probabilistic facts
 0.5::head1. 0.6::head2.

#rule
 twoHeads:-
   head1,
   head2.

#query
 query(twoHeads).

0.3
```

Logic Programming:

- **Facts**:

  unconditionally true statements on both object and their relations

- **Rules**:

  relations among objects

- **Queries**:

  interaction with the logic program

DeepProbLog

```
#neural predicate
 nn(coin_nn, [X], Y, [h,t]): coin(X, Y)

#rule
 twoHeads(X1, X2):-
   coin(X1, h),
   coin(X2, h).

#query
```

query(twoHeads(,))

0.6

An application of a NS in the context of (structured) event recognition

Introduction & Motivation

- **Events:**

Introduction & Motivation

- **Events:**

- Neural approaches:
  - Not/Limit support for background knowledge
  - Large amount of annotated training data

Introduction & Motivation

- **Events:**

- Neural approaches:
  - Not/Limit support for background knowledge
  - Large amount of annotated training data
- Neuro-symbolic approaches:
  - Support for background knowledge
  - Less amount of annotated training data → "shallow" annotations

Contribution

1 A formal definition of Structured event (Se) recognition using "shallow" annotations

2 A neuro-symbolic prototype using DeepProbLog

3 A framework to generate fully annotated videos

4 Experiment: Neural vs DeepProbLog approach

Problem definition

- A first order language $\mathcal{L}$:
  - three sorts: $\mathbb{O}$ (objects), $\mathbb{E}$ (events), $\mathbb{T}$ (time-points)
  - constants $0, 1, 2, \ldots$ of sort $\mathbb{T}$
  - $<: \mathbb{T} \times \mathbb{T} \to \{\top, \bot\}$
  - $\mathcal{P}$ of sort $\mathbb{O}^k \to \{\bot, \top\}$, $\mathcal{E}$ of sort $\mathbb{O}^k \to \mathbb{E}$
  - $\mathit{outcome}(\mathbb{E}, \mathbb{O})$, $\mathit{happens}(\mathbb{E}, \mathbb{T}, \mathbb{T})$

Problem definition

- A first order language $\mathcal{L}$:
  - three sorts: $\mathbb{O}$ (objects), $\mathbb{E}$ (events), $\mathbb{T}$ (time-points)
  - constants $0, 1, 2, \ldots$ of sort $\mathbb{T}$
  - $<: \mathbb{T} \times \mathbb{T} \rightarrow \{\top, \bot\}$
  - $\mathcal{P}$ of sort $\mathbb{O}^k \rightarrow \{\bot, \top\}$, $\mathcal{E}$ of sort $\mathbb{O}^k \rightarrow \mathbb{E}$
  - $outcome(\mathbb{E}, \mathbb{O})$, $happens(\mathbb{E}, \mathbb{T}, \mathbb{T})$
- Example of formulas:
  - $\exists x. happens(leave(John, x), t_1, t_2)$
  - $milk(x) \wedge coffee(y) \rightarrow outcome(mix(x, y), z) \wedge cappuccino(z)$

Problem definition

- **Our aim:**

  From a data sequence $\boldsymbol{D} = \{\boldsymbol{d}_i\}_{i=1}^{k}$ generate an interpretation (i.e., a description) of what happens in $\boldsymbol{D}$

Problem definition

- **Our aim:**

  From a data sequence $\boldsymbol{D} = \{\boldsymbol{d}_i\}_{i=1}^{k}$ generate an interpretation (i.e., a description) of what happens in $\boldsymbol{D}$

- Example:

  $\boldsymbol{D} = \{\boldsymbol{d}_i\}_{i=1}^{7} \rightarrow$ person moves, leaves a bag and moves again:

Problem definition

- **Our aim:**

  From a data sequence $\boldsymbol{D} = \{\boldsymbol{d}_i\}_{i=1}^{k}$ generate an interpretation (i.e., a description) of what happens in $\boldsymbol{D}$

- Example:

  $\boldsymbol{D} = \{\boldsymbol{d}_i\}_{i=1}^{7} \rightarrow$ person moves, leaves a bag and moves again:

  - $\mathcal{C} = \{p_1, b_1\}$

  - $$\mathcal{F} = \left\{ \begin{array}{c} person(p_1), bag(b_1), \\ happens(move(p_1), 0, 4), happens(leave(p_1, b_1), 4, 5) \\ happens(move(p_1), 5, 7), \end{array} \right\}$$

Problem definition

- **Our aim:**

  From a data sequence $\boldsymbol{D} = \{\boldsymbol{d}_i\}_{i=1}^{k}$ generate an interpretation (i.e., a description) of what happens in $\boldsymbol{D}$

- Example:

  $\boldsymbol{D} = \{\boldsymbol{d}_i\}_{i=1}^{7} \rightarrow$ person moves, leaves a bag and moves again:

  - $\mathcal{C} = \{p_1, b_1\}$

  - $$\mathcal{F} = \left\{ \begin{array}{c} person(p_1), bag(b_1), \\ happens(move(p_1), 0, 4), happens(leave(p_1, b_1), 4, 5) \\ happens(move(p_1), 5, 7), \end{array} \right\}$$

- Partial supervision:

  $\rightarrow$ subset of events that happened (or do not) in $\boldsymbol{D}$

Problem definition

- Example(cont.):

$$
\begin{aligned}
& \mathit{happens}(\mathit{potential\_threat}, t_0, t_3) \leftrightarrow \\
& \qquad \exists x, y, t_1, t_2.\ \mathit{person}(x) \wedge \mathit{bag}(y) \wedge \\
& \qquad\qquad \mathit{happens}(\mathit{move}(x), t_0, t_1) \wedge \\
& \qquad\qquad \mathit{happens}(\mathit{leave}(x, y), t_1, t_2) \wedge \\
& \qquad\qquad \mathit{happens}(\mathit{move}(x), t_2, t_3)
\end{aligned}
$$

Problem definition

- Example(cont.):

$$
\begin{aligned}
&\textit{happens}(\textit{potential\_threat}, t_0, t_3) \leftrightarrow \\
&\qquad \exists x, y, t_1, t_2.\ \textit{person}(x) \land \textit{bag}(y) \land \\
&\qquad\qquad \textit{happens}(\textit{move}(x), t_0, t_1) \land \\
&\qquad\qquad \textit{happens}(\textit{leave}(x, y), t_1, t_2) \land \\
&\qquad\qquad \textit{happens}(\textit{move}(x), t_2, t_3)
\end{aligned}
$$

- Partial Supervision:
  - $\boldsymbol{D}^{(1)}, \boldsymbol{D}^{(2)}, \ldots, \boldsymbol{D}^{(m)}$: m videos of length k
  - $\boldsymbol{D}^{(i)}$:
    - $\rightarrow$ $\textit{happens}(\textit{potential\_threat}, 0, k)$
    - $\rightarrow$ $\neg\textit{happens}(\textit{potential\_threat}, 0, k)$

Proposed solution

- Three tasks has to be solved:
  1. Object detection
  2. Object classification and relation detection
  3. Event recognition

Proposed solution

- Three tasks has to be solved:
  1. Object detection
  2. Object classification and relation detection
  3. Event recognition
- Use neural networks:
  - A ntw for detecting the objects (Det_{nn})
  - A ntw for predicate evaluation (P_{nn})

Proposed solution

- Three tasks has to be solved:
    1. Object detection
    2. Object classification and relation detection
    3. Event recognition
- Use neural networks:
    - A ntw for detecting the objects (Det_{nn})
    - A ntw for predicate evaluation (P_{nn})
- Combines ntws outputs with background knowledge:
    - → DeepProbLog prototype

Event Generation Framework

- Different level of annotations

Event Generation Framework

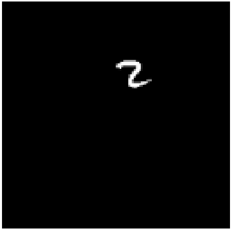
- Different level of annotations
- Manually curated and not extensible

Event Generation Framework

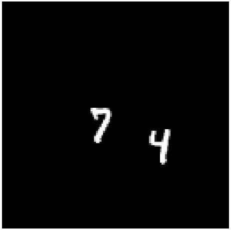
- Different level of annotations
- Manually curated and not extensible
- Mnist digits video generator:
  - video with different length and different number of objects
  - simple events (appear, disapper, enter and exit)
  - structured events (join_add, join_sub and split)
  - narrative (name, class, position, simple events and structured events)

Event Generation Framework

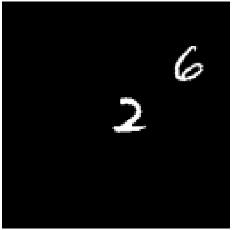
Join_add



Join_sub



Split



Experiments

Experimental setting

- **Research question:**

  Has a neuro-symbolic solution an advantage in recognizing Se with respect to a fully neural approach?

Experimental setting

- **Research question:**

  Has a neuro-symbolic solution an advantage in recognizing Se with respect to a fully neural approach?

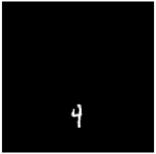
- Evaluation:
  - correct classification of the video
  - correct classification of the objects (i.e. digits)
  - generalization to unseen outcomes (i.e. no explicit supervision)

Experimental setting

Scenario:

- videos of 10 frames
- digits appear anytime within the first half of the video, and only disappear if they join together
- three events:

Join_add



Join_sub

No_join

Experimental setting

Learning:

$$\{happens(join\_add(x, y), 1, T), outcome(join\_add(x, y), z), digit(z, 4)\}$$

$$\{happens(join\_sub(x, y), 1, T), outcome(join\_sub(x, y), z), digit(z, 2)\}$$

$$\{\neg happens(join\_add(x, y), 1, T), \neg happens(join\_sub(x, y), 1, T)\}$$

Experimental setting

Learning:

$$\{happens(join\_add(x,y),1,T), outcome(join\_add(x,y),z), digit(z,4)\}$$

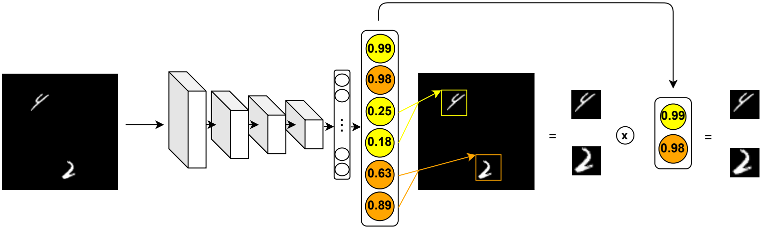
$$\{happens(join\_sub(x,y),1,T), outcome(join\_sub(x,y),z), digit(z,2)\}$$

$$\{\neg happens(join\_add(x,y),1,T), \neg happens(join\_sub(x,y),1,T)\}$$

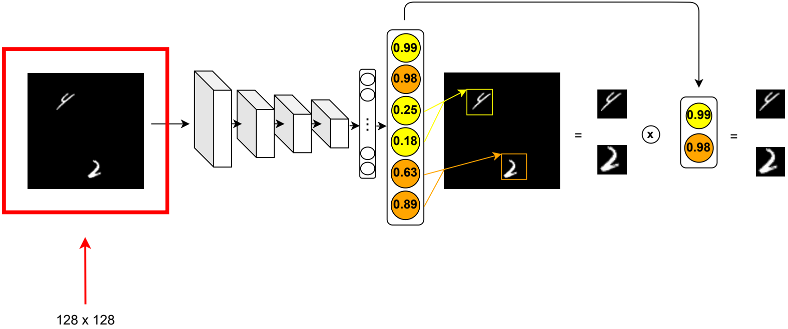
- train & validation (1800 and 150 videos):
  - → join_add – outcome from 2 to 7
  - → join_sub – outcome from 0 to 7
  - → no_join
- test (180 videos):
  - → join_add – outcome from 2 to 9
  - → join_sub – outcome from 0 to 8
  - → no_join

Event recognition approaches

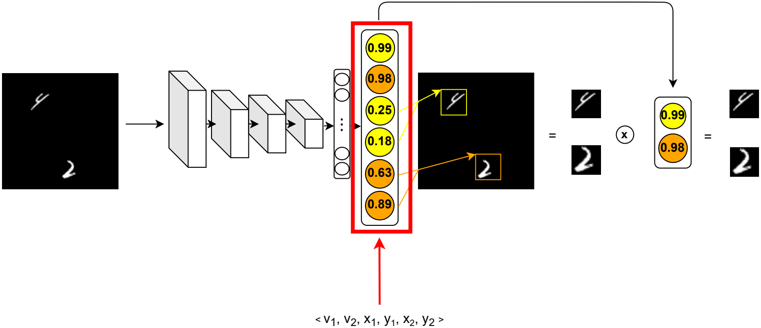
Object detector and classifier



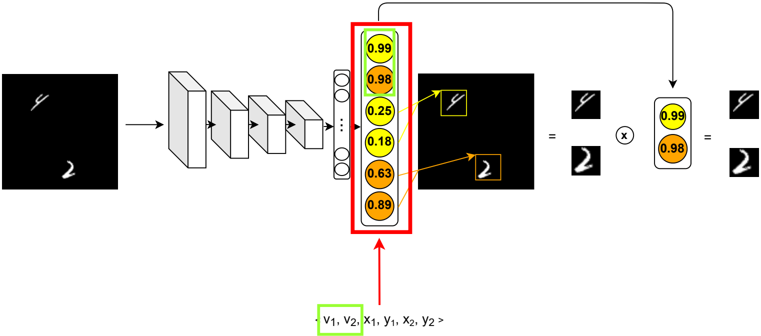
Object detector and classifier



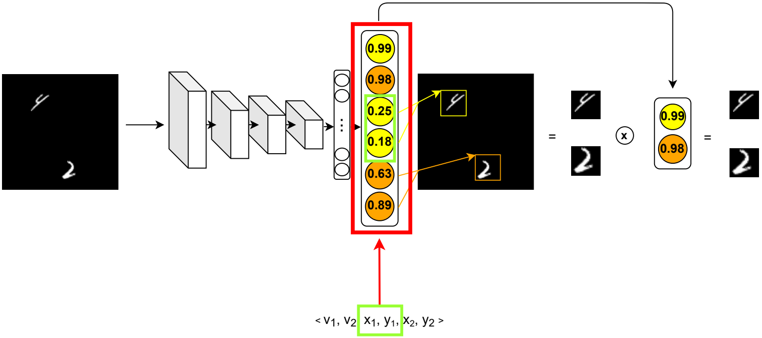
Object detector and classifier



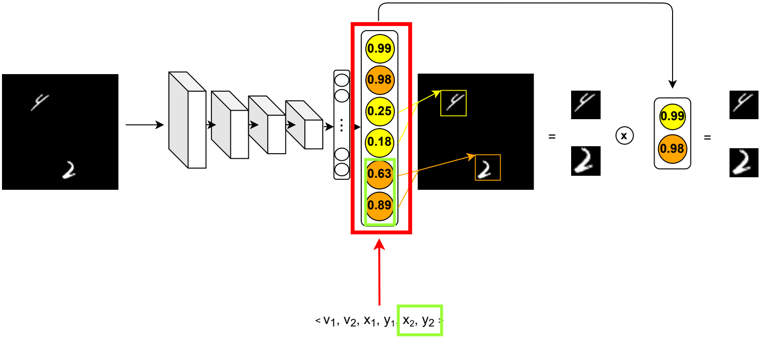
Object detector and classifier



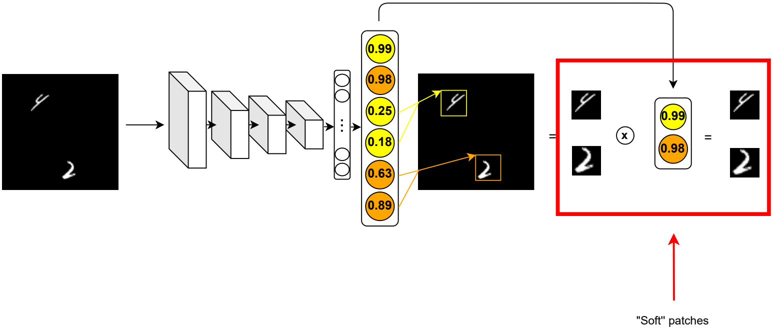
Object detector and classifier



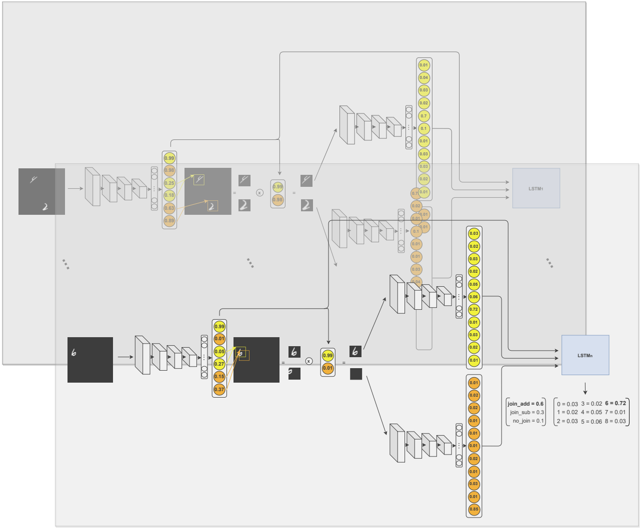
Object detector and classifier



Object detector and classifier



Fully neural approach



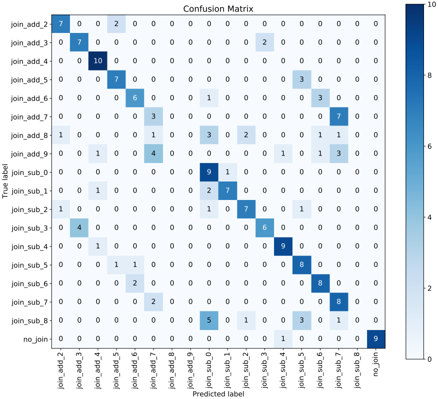
Neuro symbolic approach

```
nn(mnist_net, [I, V, T], Y, [0,1,2,3,4,5,6,7,8,9,-1]) :: digit(I, V, T, Y).

join_add_res(V, Z) :-
    between(0, 4, T1),
    digit(0, V, T1, X),
    X > 0, X < 9,
    digit(1, V, T1, Y),
    Y > 0, Y < 10 - X,
    digit(0, V, 9, Z),
    Z is X + Y, Z > 1,
    digit(1, V, 9, -1).

join_sub_res(V, Z) :-
    between(0, 4, T1),
    digit(0, V, T1, X),
    X > 0,
    digit(1, V, T1, Y),
    Y > 0,
    digit(0, V, 9, Z),
    Z is abs(X-Y),
    digit(1, V, 9, -1).

no_join(V) :- digit(1, V, 9, X), X =\= -1.
```

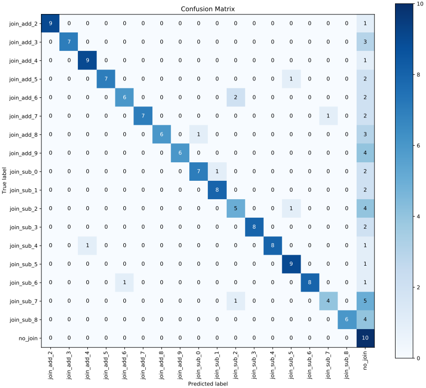
Results

Results - Structured event recognition

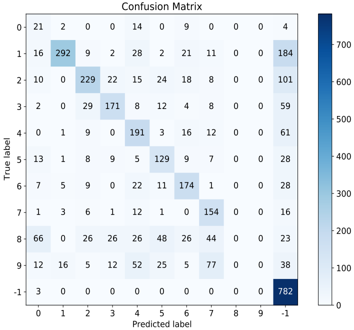
Neural approach



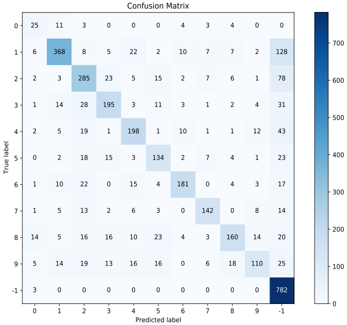
DeepProbLog



Neural approach



DeepProbLog



Conclusion

Summary:

- neuro-symbolic approach based on DeepProbLog for Se recognition
- end-to-end training using shallow annotations
- comparison with pure neural approach:
    1) train without direct supervision on some classes
    2) explainability

Thank you!