

# A Deep Learning based Framework for UAV Trajectory Pattern Recognition

**Xingyu PAN**

Pascal DESBARATS

Serge CHAUMETTE



# Outline

- **Context**
- **Data preparation**
- **Deep learning methods**
- **Experiments and results**
- **Conclusion and future work**

# Context

## ■ Motivation

- 3D environment perception by autonomous system (swarm of drones)
- Intelligent supervision of UAVs for public and military security
  - Mission recognition
  - Abnormal behaviors supervision
  - UAV & non-UAV object detection

## ■ Similar study:

**SafeShore**

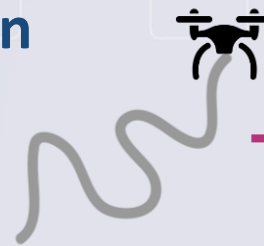
Workshop AVSS 2017

Object detection & shape analysis  
based on CNN (Coluccia, *et al.* 2017)



# Context

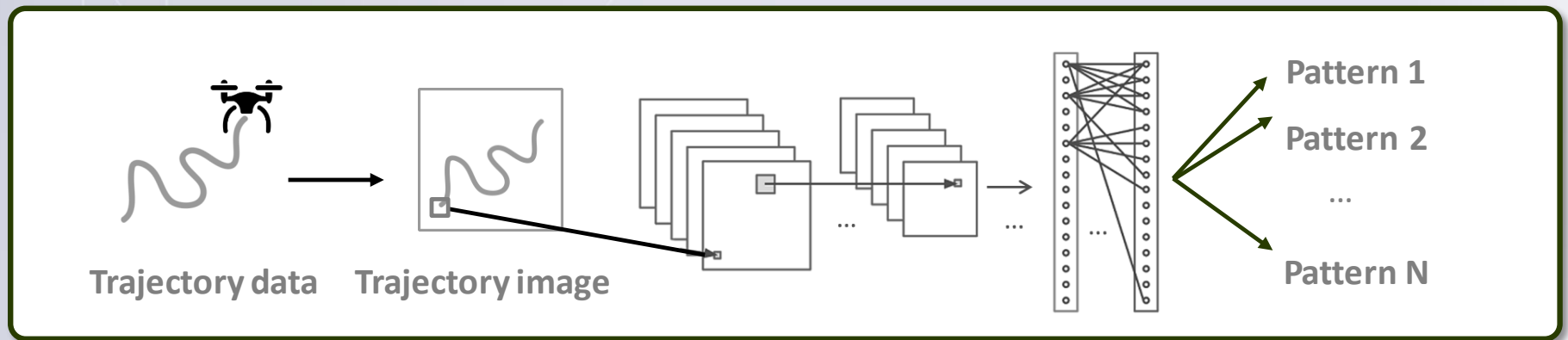
## UAV Trajectory pattern recognition



Drone?  
Mission?  
Normal behavior?

## Convolutional neural networks

(Krizhevsky *et al.* 2012, Szegedy *et al.* 2015, Simonyan and Zisserman 2015, He *et al.* 2016)



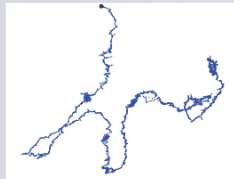
Trajectory data → Trajectory image

Trajectory pattern recognition → image classification

# Context

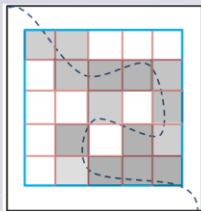
## ■ Problems & references

### ■ Trajectory images of moving objects



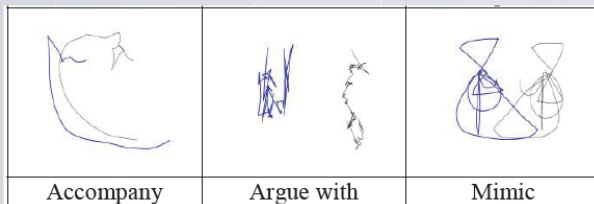
**Migrant animals** (Wang 2016)

Simple spatial trajectories



**Transportation modes** (Endo *et al.* 2016)

Trajectories with grid intensity according to speed



**Pretend game players** (Singh *et al.* 2016)

Multi-object trajectories

# Context

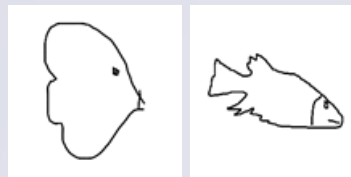
## ■ Problems & references

### ■ Sketch-like data recognition

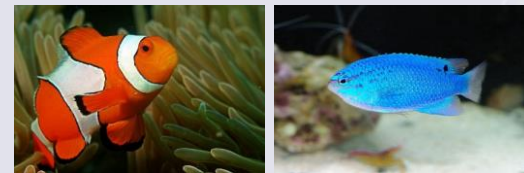
#### ■ Sketch databases:

*TU-Berlin* (Elitz *et al.* 2012)

*Sketchy* (Sankoly *et al.* 2016)



Sketches

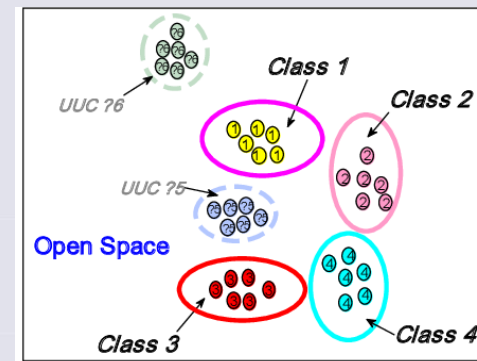


Photos

### ■ Open set recognition

#### ■ 1-vs-rest SVM

#### ■ Extension of softmax (Bendale et E. Boult 2016)



\*Data acquisition & preprocessing considered to be ideal

# Outline

- Context
- **Data preparation**
- Deep learning methods
- Experiments and results
- Conclusion and future work

# Data preparation

- Lack of real labeled data
- Synthetic data generation using *dh* **DRONE HARMONY**
  - Synthetic 3D scenes with real GPS coordinates
  - Predefined **missions**
  - Automatic trajectory simulation

OpenStreetMap

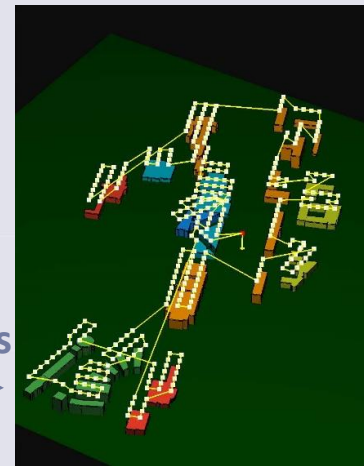


Create target  
Buildings / terrains



Scene 1: Talence campus

Set **mission**  
Set parameters



Export  
waypoints (.csv)

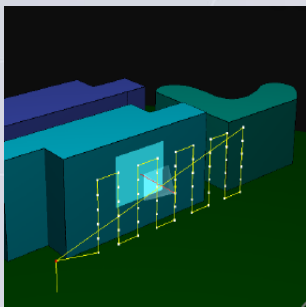


# Data preparation

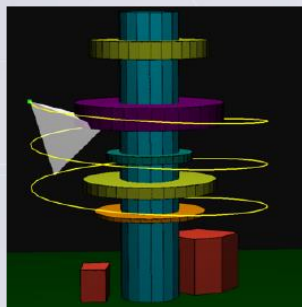
- 2 scenes: Talence campus, Bordeaux city center
- 5 classes (predefined missions)
  - Façade inspection (fcd)
  - Tower inspection (hlx)
  - Orbit mapping (obm)
  - Side inspection (per)
  - Top-down mapping (tdm)



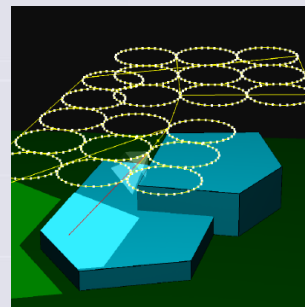
Scene 2: Bordeaux city center



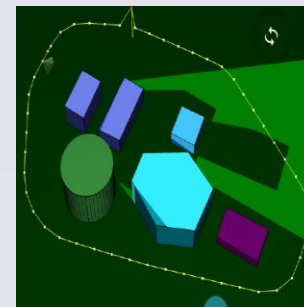
fcd



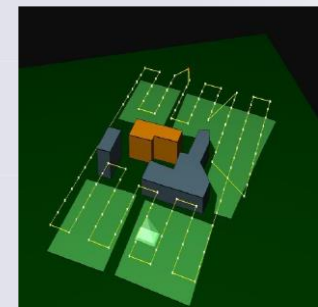
hlx



obm



per



tdm

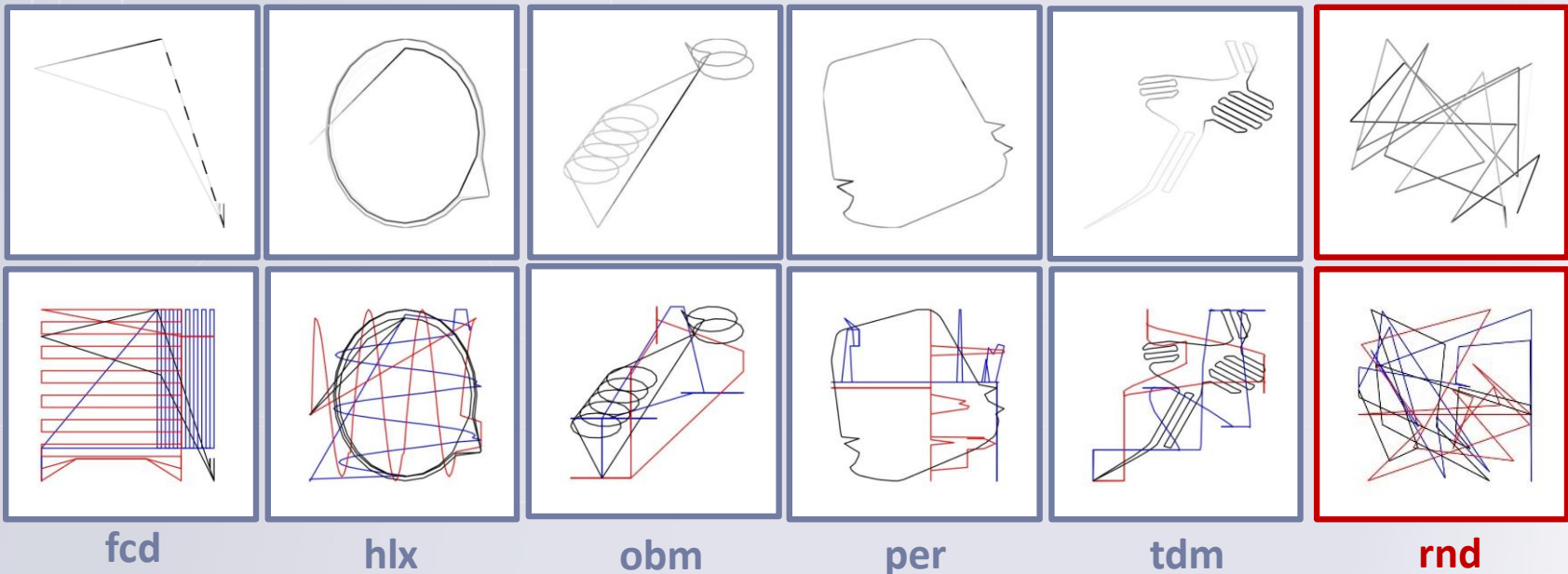
# Data preparation

## 3D GPS waypoints → 2D Trajectory images

- 1-channeled XY images (grayscale, intensity according to altitude)
- 3-channeled XYZ images (color, superposed 3 planes)

Data of known classes

Random unknow data



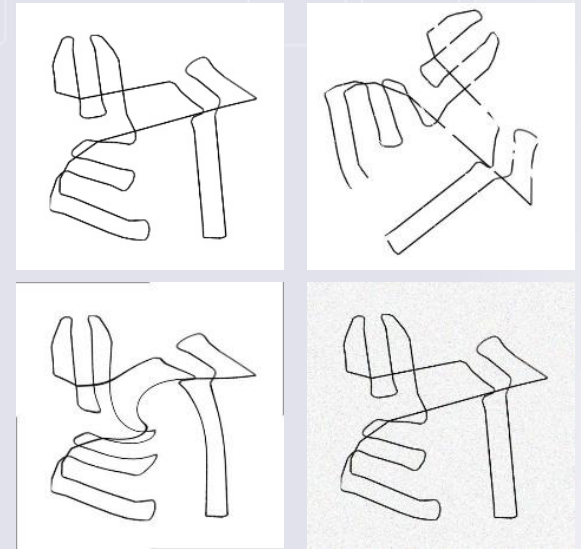
# Data preparation

## ■ Data augmentation

- Rotation
- Horizontal / vertical flip
- Background noise
- Swirl distortion
- Random erasing

## ■ Database simTrj (simulated trajectories)

- 6 classes (including random unknown trajectories)
- 3908 images
  - 1954 XY-images
  - 1954 XYZ-images



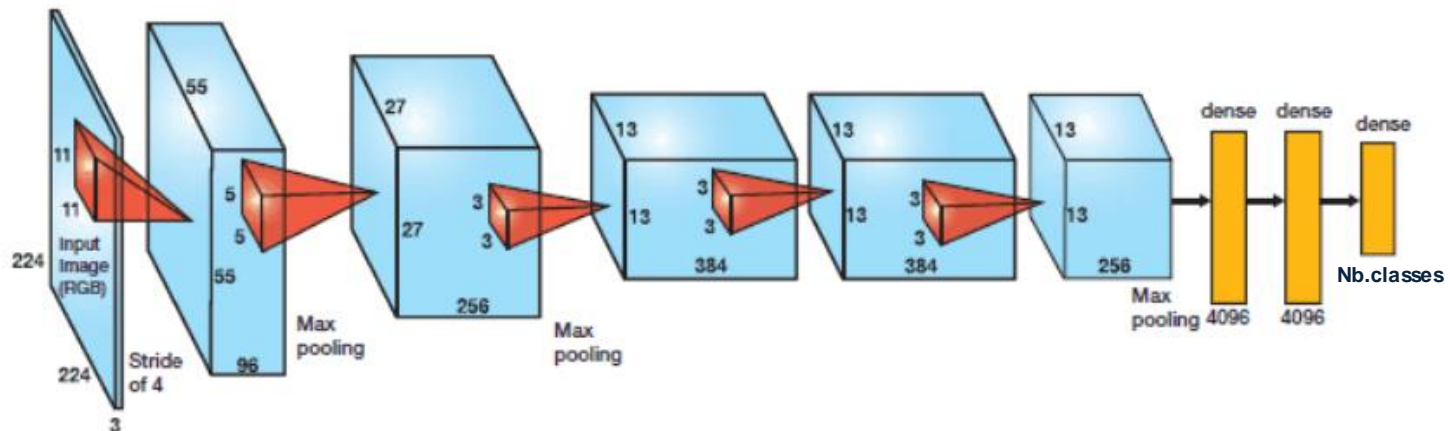
# Outline

- Context
- Data preparation
- **Deep learning methods**
- Experiments and results
- Conclusion and future work

# Deep learning methods

- Deep network architecture
  - Performant & light network
    - Small database
    - Hardware constraint (1 GPU)

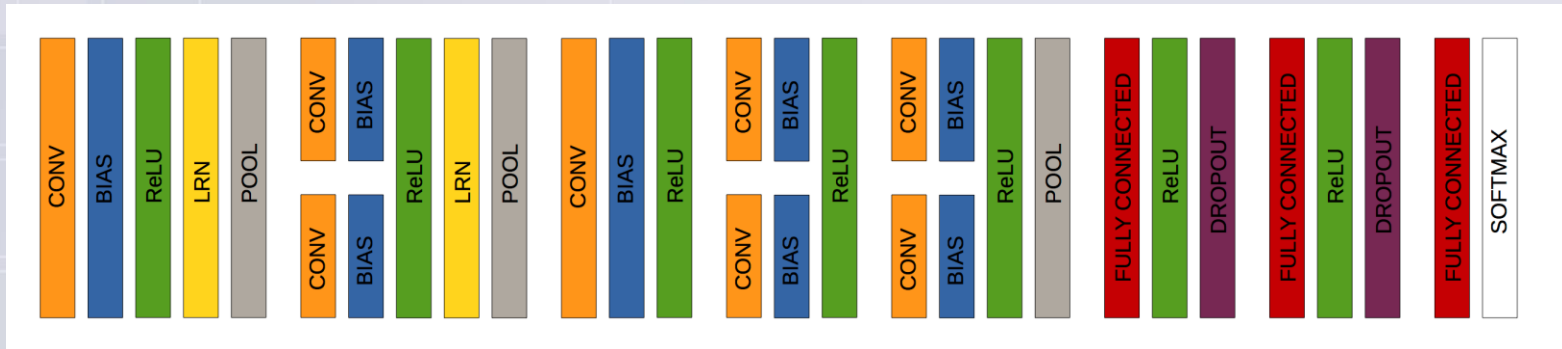
**AlexNet** (Krizhevsky *et al.* 2012)



# Deep learning methods

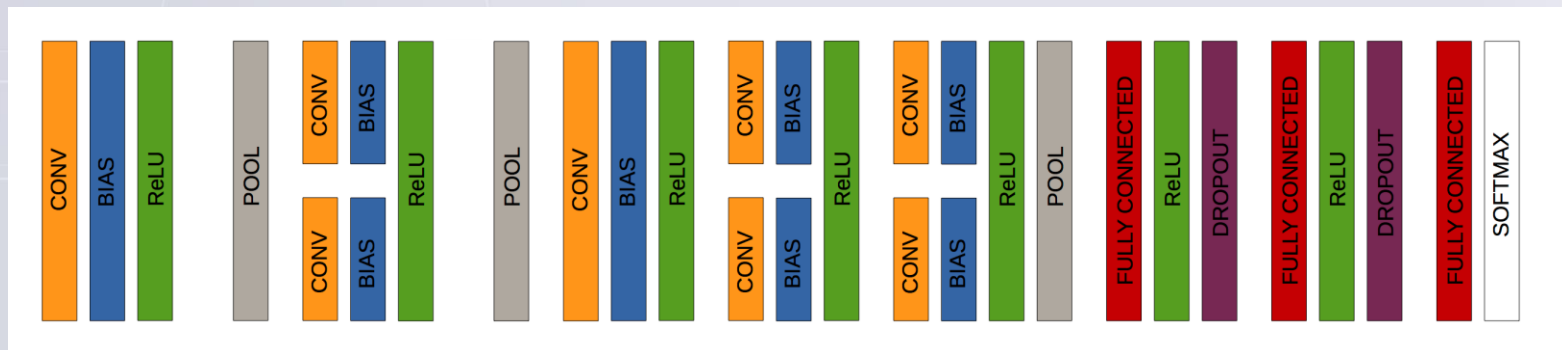
## Deep framework: Caffe (GPU acceleration)

### AlexNet



### SkalexNet (Adaption of AlexNet to sketches)

- Remove Local Response Normalization (LRN) layers



# Deep learning methods

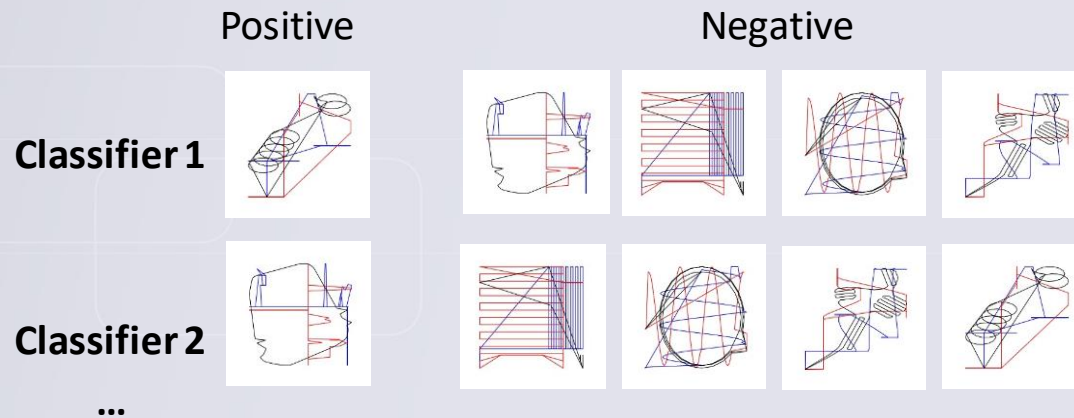
- Unknown trajectories

- N+1 multiclass classification

- Train and test **unknown trajectories** as a new class (rnd)

- 1-vs-rest classification

- N binary classifiers trained by **known trajectories**
    - Only test data including **unknown trajectories**



# Outline

- Context
- Data preparation
- Deep learning methods
- Experiments and results
- Conclusion and future work



# Experiments and results

## Experiments

### 4 subsets of *simTrj* database

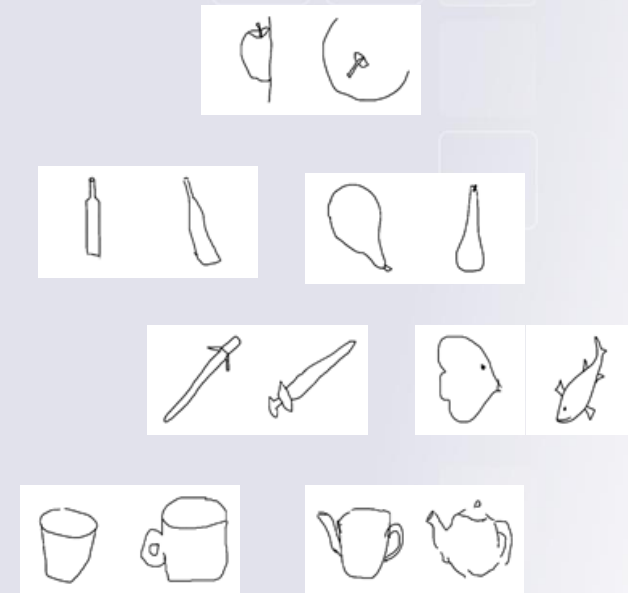
- *simTrj2gray*: 1604 XY-images, 5 classes
- *simTrj2color*: 1604 XYZ-images, 5 classes
- *simTrj3gray*: 1954 XY-images, 6 classes
- *simTrj3color*: 1954 XYZ-images, 6 classes

### *Sketchy0*

- preselection of *Sketchy* database (Sankoly et al. 2016)
- 4184 trajectory-like sketches

### Convolutional networks (AlexNet & SkalexNet)

- Training from scratch
- Classic layer settings
- Experimental hyperparameters



*Sketchy0*

# Experiments and results

## Results

### Multiclass classification

| Database     | N. class | N.train | N.test | AlexNet | SkalexNet |
|--------------|----------|---------|--------|---------|-----------|
| SimTrj2gray  | 5        | 1000    | 604    | 98.5%   | 98.8%     |
| SimTrj2color | 5        | 1000    | 604    | 99.2%   | 99.5%     |
| SimTrj3gray  | 6        | 1200    | 754    | 97.8%   | 97.5%     |
| SimTrj3color | 6        | 1200    | 754    | 98.4%   | 98.6%     |
| Sketchy0     | 7        | 3500    | 684    | 98.3%   | 98.9%     |

XYZ-images > XY-images (slightly)

SkalexNet > AlexNet (slightly)

### 1-vs-rest classification (*simTrj3color*)

200 positives vs. 200 negatives for each classifier training

UAV vs. non-UAV detection:

Overall accuracy: 95.4%, UAV accuracy: 97.6%, **Precision: 99.5%, recall: 85.6%**

# Experiments and results

## ■ Confusion matrix

### ■ Multiclass classification (*simTrj3color*)

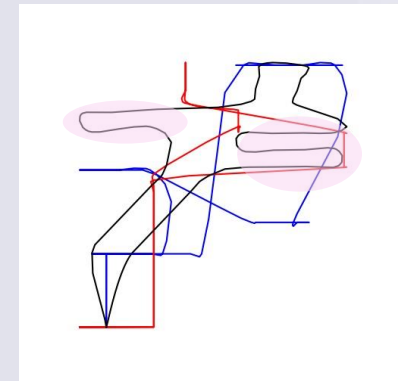
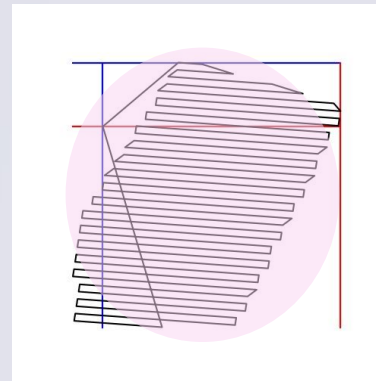
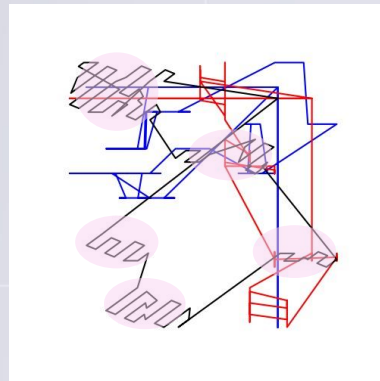
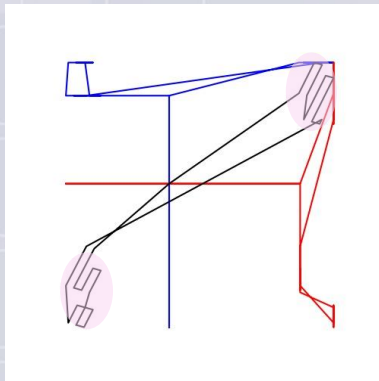
|     | fcd     | hlx     | obm     | per      | tdm     | rnd     |
|-----|---------|---------|---------|----------|---------|---------|
| fcd | 99,06 % | 0,00 %  | 0,00 %  | 0,00 %   | 0,94 %  | 0,00 %  |
| hlx | 0,00 %  | 99,09 % | 0,00 %  | 0,00 %   | 0,91 %  | 0,00 %  |
| obm | 0,87 %  | 0,00 %  | 94,78 % | 2,61 %   | 0,87 %  | 0,87 %  |
| per | 0,00 %  | 0,00 %  | 0,00 %  | 100,00 % | 0,00 %  | 0,00 %  |
| tdm | 2,19 %  | 0,00 %  | 0,73 %  | 0,00 %   | 97,08 % | 0,00 %  |
| rnd | 1,33 %  | 0,00 %  | 0,67 %  | 0,00 %   | 0,00 %  | 98,00 % |

### ■ 1-vs-rest classification (*simTrj3color*)

|     | fcd     | hlx      | obm     | per     | tdm      | rnd     |
|-----|---------|----------|---------|---------|----------|---------|
| fcd | 98,67 % | 0,00 %   | 0,00 %  | 0,00 %  | 0,67 %   | 0,67 %  |
| hlx | 0,00 %  | 100,00 % | 0,00 %  | 0,00 %  | 0,00 %   | 0,00 %  |
| obm | 0,00 %  | 0,00 %   | 94,67 % | 0,00 %  | 4,67 %   | 0,67 %  |
| per | 0,67 %  | 0,00 %   | 0,00 %  | 94,67 % | 1,33 %   | 3,33 %  |
| tdm | 0,00 %  | 0,00 %   | 0,00 %  | 0,00 %  | 100,00 % | 0,00 %  |
| rnd | 0,29 %  | 0,00 %   | 0,00 %  | 0,00 %  | 36,86 %  | 62,86 % |

# Experiments and results

- Most confusing class: top-down mapping (tdm)
- Great global intraclass variability



- Class tdm featured by XY-undulation patterns
- Undulation location according to distribution of targets
- Undulation density according to relative altitude

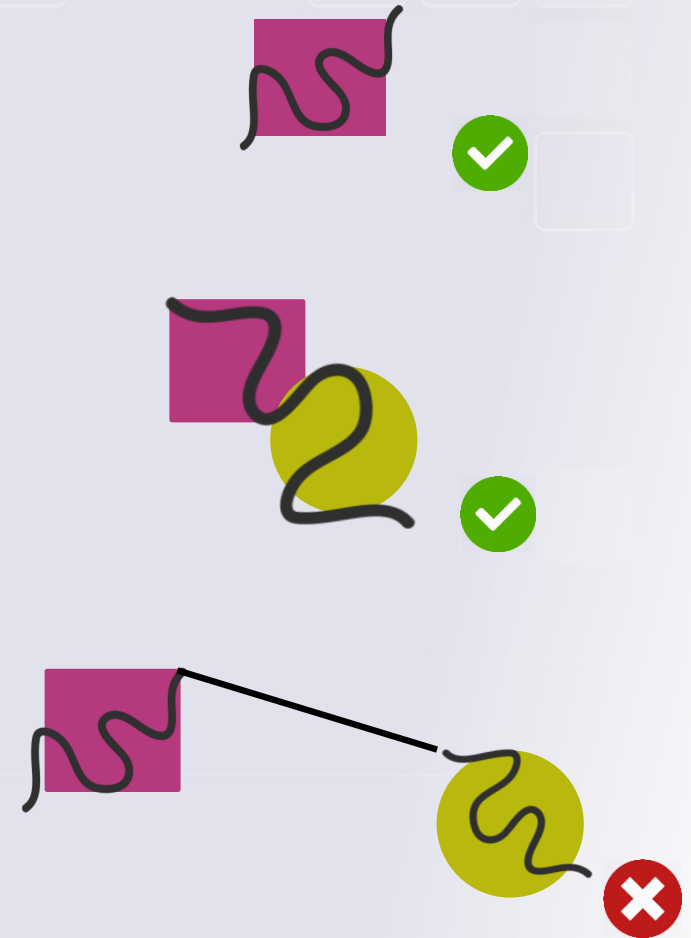
# Experiments and results

## Supplementary experiments

- One-target inspection
  - Mission on one target
  - No trajectories between targets

| Databases        | <i>AlexNet</i> | <i>SkalexNet</i> |
|------------------|----------------|------------------|
| <i>simTrj2OT</i> | 100.0%         | 100.0%           |
| <i>simTrj3OT</i> | 99.5%          | 100.0%           |

\**simTrj2OT*: one-target XY-trajectory images  
*simTrj3OT*: one-target XYZ-trajectory images

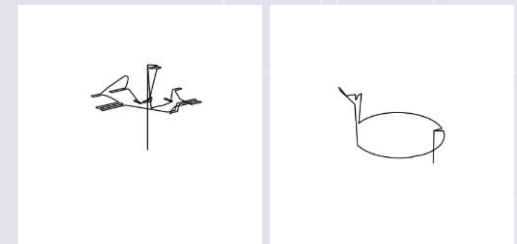


# Experiments and results

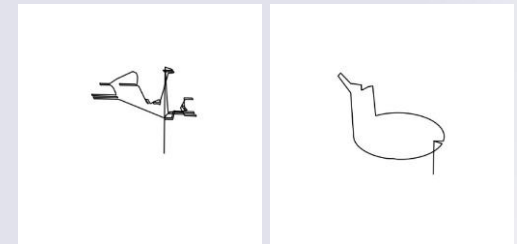
## ■ Supplementary experiments

- One camera acquisition
  - 2D view from a fixed angle
  - Not perpendicular to XY, YZ or XZ plans

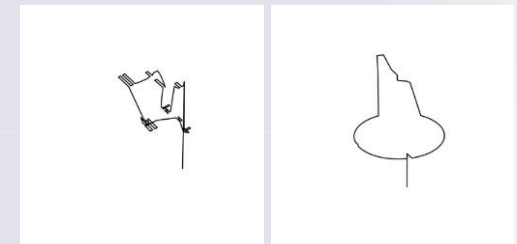
| Databases           | AlexNet | SkalexNet |
|---------------------|---------|-----------|
| <i>simTrj3d1</i>    | 96.7%   | 96.7%     |
| <i>simTrj3d2</i>    | 98.7%   | 98.2%     |
| <i>simTrj3d3</i>    | 97.6%   | 97.9%     |
| <i>simTrj3color</i> | 98.4%   | 98.6%     |



SimTrj3d1 (30%)



SimTrj3d2(45%)



SimTrj3d3(60%)

\**simTrj3dx*: trajectory 2D projections with a certain angle

# Outline

- Context
- Data preparation
- Deep learning methods
- Experiments and results
- Conclusion and future work

# Conclusion and future work

## ■ Contributions

- First work on UAV mission recognition based on trajectories
- Synthetic UAV trajectory database
- Adapted and light deep network for sketch-like data
  - AlexNet based architecture
  - Removing LRN increases speed & accuracy

X. Pan, P. Desbarats, S. Chaumette, *A Deep Learning Framework for UAV Trajectory Pattern Recognition*, IPTA 2019, Istanbul



# Conclusion and future work

## ■ Future work

- Database extension
  - More simulated trajectories
  - More classes
  - New image representation
  - Real database acquisition
- More complicated deep networks
  - Deeper networks
  - Implementation of open set recognition on output layer
- Trajectory prediction
- Multi-trajectories for MAS (Multiple Agent System)



**Thank you!**