IMT Lille Douai
École Mines-Télécom
IMT-Université de Lille

IFSTTAR

Convolutional Neural Network based video analysis for road users detection and interactions modeling

PHD STUDENT: VU TUAN HUNG

SUPERVISOR: JACQUES BOONAERT (IMT LILLE DOUAI)

SEBASTIEN AMBELLOUIS (IFSTTAR)

ABDELMALIK TALEB-AHMED (UNIV. VALENCIENNES)

Outline

1. Introduction
2. State of the art of CNN models
3. Instance segmentation with Mask RCNN
4. Improved tracking by predicting future segmentation
5. Future works
6. Conclusions

1. Introduction

European H2020 ORIO project

European H2020 PROSPECT project

❑CONTEXT

➢**Scenario:** Vulnerable Road Users (pedestrian, bike, scooter etc.) and their interactions with truck/car/motobikes

➢**Scientific context**:

- Road users detection and tracking, dense trajectories estimation, light condition change, crowdy situation
- Action recognition & abnormal activities detection, simultaneous actions

1. Introduction

❑ OBJECTIVES

➢ Practical application:

- **Detect**, **segment** and **track VRU** and **transport infrastructure** in video surveillance to **recognize particular**, **critical** or **dangerous activities**: car moving too fast or too close to a pedestrian, people crossing outside zebra, etc.

➔ ***Visual action recognition & understanding problem***

➢ Traditional pipeline:

- Constructing appropriate models for **extracting efficient representation of information** in video (e.g: HOG, SIFT, SURF, STIP, etc)
- Applying **robust learning model** (e.g: Perceptron, Regression, SVM, kNN, etc)

1. Introduction

❑ Visual action recognition and understanding

➢ Why not traditional methods ?

- Traditional context: daily & normal actions**,** single actions, simple context
- Our context: multi-task, simultaneous actions, crowdy situation, abnormal activities

➔ DEEP LEARNING

➢ Scientific solution:

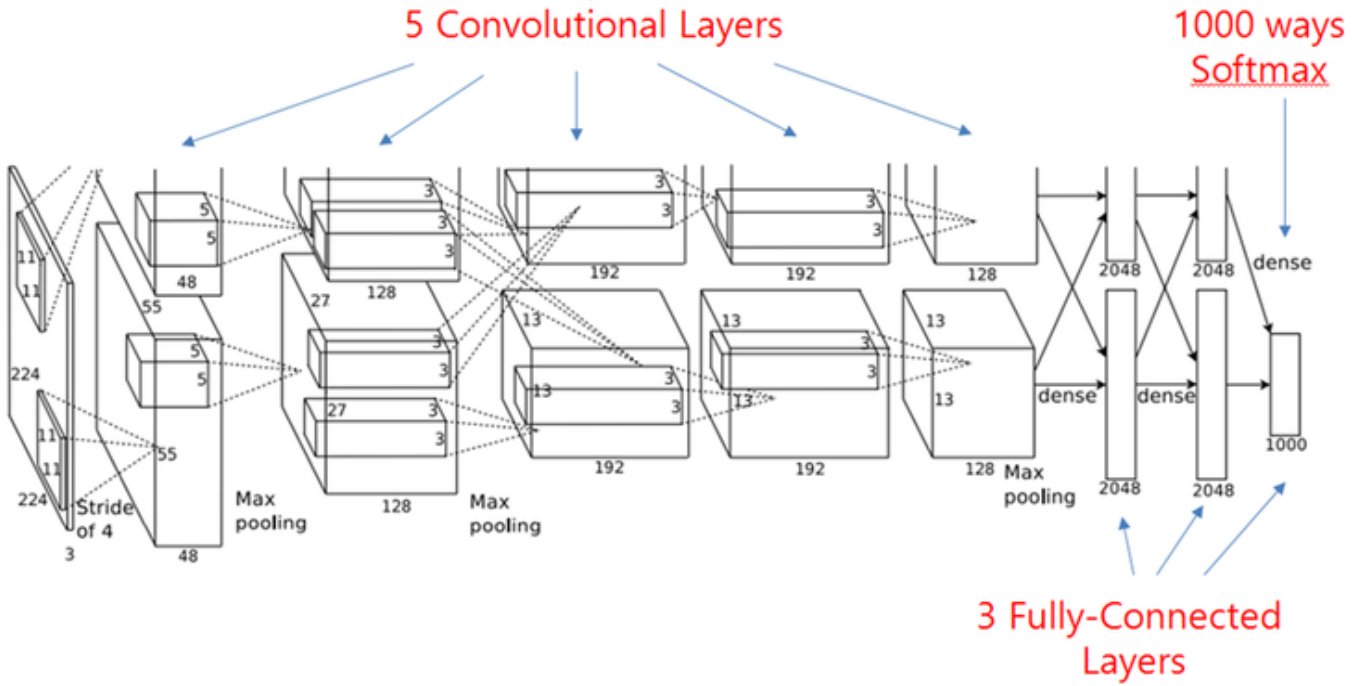
- Combine state-of-the-art **Deep learning** methods with specific techniques to **improve performance of object detection**, **segmentation** and **tracking**
- Propose **new efficient features** or a **new network structure** and construct effective learning models for our particular activities recognition.

Outline

1. Introduction
2. State of the art of CNN models
3. Instance segmentation with Mask RCNN
4. Improved tracking by predicting future segmentation
5. Future works
6. Conclusions

2. CNN models

- Convolutional Neural Network (CNN)

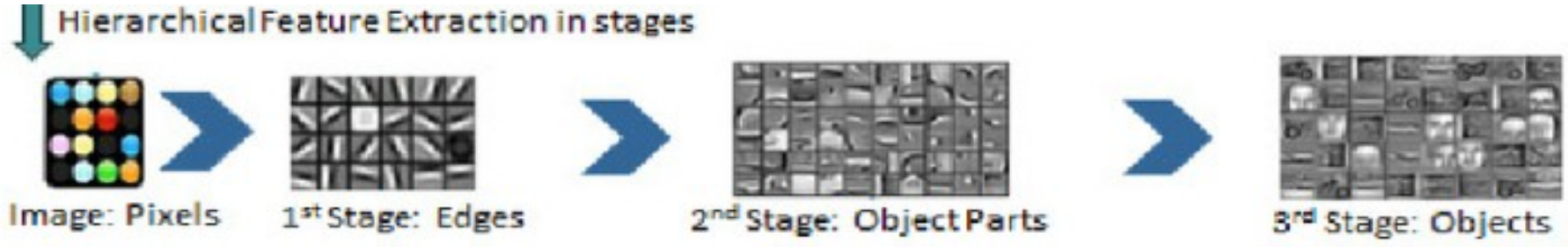


AlexNet architecture

2. CNN models

Convolutional Neural Network (CNN)

- The deeper the layer, the more semantic the information
- All filters parameters are updated at each iteration during the training task ➔ CNNs features are more adaptive to learning task

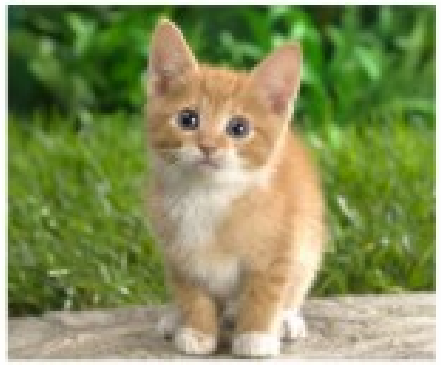


2. CNN models

- Large review of the state-of-the-art related to our problem
  - Learning techniques: unsupervised (GANs architecture), supervised (ResNet architecture)
  - Images classification: AlexNet, GoogLeNet, VGG-Net, **ResNet**
  - Action classification: Two-stream CNNs, Trajectories-Pooled Deep Descriptor
  - Object detection: R-CNN, SPP-Net, Fast RCNN, **Faster RCNN**, **YOLO**
  - Segmentation: Fully Convolutional Network (FCN), SegNet, **DeepLab**, RefineNet, **Mask RCNN**
  - Optical flow estimation: DeepFlow, EpicFlow, **FlowNet**
  - Moving object segmentation: **MP-Net**

2. CNN models

- State-of-the-art:
  - Image classification



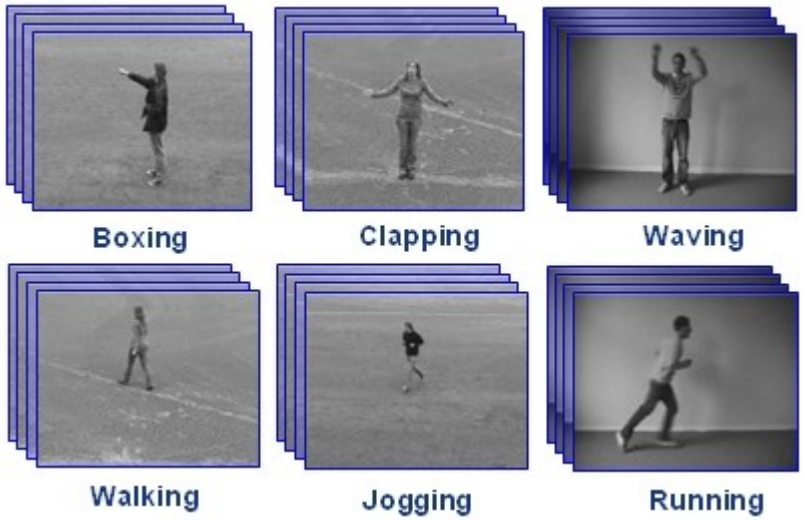
CAT

Method	Description	Performance
AlexNET 2012 [32]	Image classification, single image, end-to-end architecture, 7 layer	17% top-5 err on ILSVRC12; 15.3% top-5 err on ILSVRC10
VGG 2014 [56]	Image Classification, single image, end-to-end architecture, 16 or 19 layers	8.43% top-5 err on ILSVRC14
GoogLeNet 2014 [58]	Image Classification, single image, end-to-end architecture, inception module, 22 layer	7.89% top-5 err on ILSVRC14

Top-5 error rate is the fraction of test images for which the correct label is not among the 5 labels considered as the most probable by the mode.

2. CNN models

- State-of-the-art:
  - Action recognition

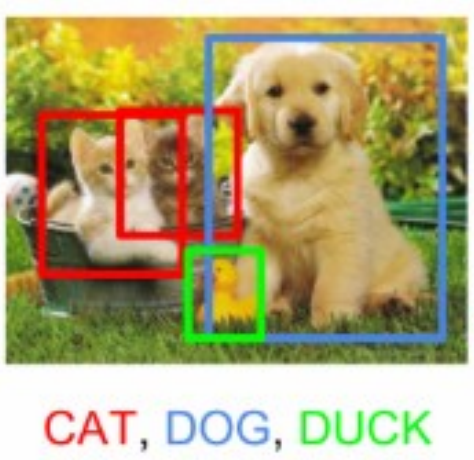


Method	Description	Performance
Dense trajectories 2012 (DT) [62]	Action Recognition, video level	ACC 83.5% on UCF101, 58.2% on Hollywood2, 46.6% on HMDB51
Improved DT 2013 (IDT) [63]	Action Recognition, video level	ACC 85.9% on UCF101, 66.8% on Hollywood2, 60.1% on HMDB51
2-streams CNNs 2014 [55]	Action Recognition, video level	ACC 88.0% on UCF101, 59.4% on HMDB51
Convolutional 3D 2014 [60]	Action Recognition, video level	ACC 85.2% on UCF101

Accuracy (ACC) is the fraction between true prediction (both true positive and true negative) and total prediction.

2. CNN models

- State-of-the-art:
  - Object detection



Method	Description	Performance
R-CNN 2013 [17]	Object Detection, AlexNet architecture, multi stages pipeline	62.4% mAP PASCAL VOC12
Fast R-CNN 2015[16]	Object Detection, end-to-end, VGG-16 architecture	68.4% mAP PASCAL VOC12, test time 0.5 fps on GPU Titan X
Faster R-CNN 2015 [42]	Object Detection, end-to-end, ResNet architecture	73.8% mAP PASCAL VOC12, test time 5 fps on GPU Titan X

Average precision (AP) computes the average value of precision over the interval of recall from 0 to 1. Mean average precision (mAP) for a set of queries is the mean of the average precision scores for each query.

2. CNN models

- State-of-the-art:
  - Object segmentation

CAT, DOG, DUCK

Method	Description	Performance
FCN 2014 [39]	Image Segmentation, sematic level	65.3% mIOU Cityscapes
SegNet 2015 [1]	Image Segmentation, semantic level	57.0% mIOU Cityscapes
DeepLab 2016 [9]	Image Segmentation, semantic level	70.4% mIOU Cityscapes
MP-Net 2017 [59]	Moving Object Segmentation, only segment moving object, first successful	69.7% mIOU DAVIS

Intersection over Union (IOU) score for each class is the fraction between true positive pixels and the sum of true positive, false negative and false positive pixels. The mean IoU (mIOU) is the mean of IOU for all classes. Especially, for instance-level segmentation, performance on this task is measured by the COCO-style mask AP (average precision on region level) and AP50 (average precision when overlap at region level is at least 50%)

2. CNN models

❑Optical Flow

➢**Optical flow**: Apparent motion of brightness patterns in the image. **Motion field:** the projection of the 3D scene motion into the image.

➔ ***Ideally, Optical Flow = Motion Field***

Brightness Constancy Equation:

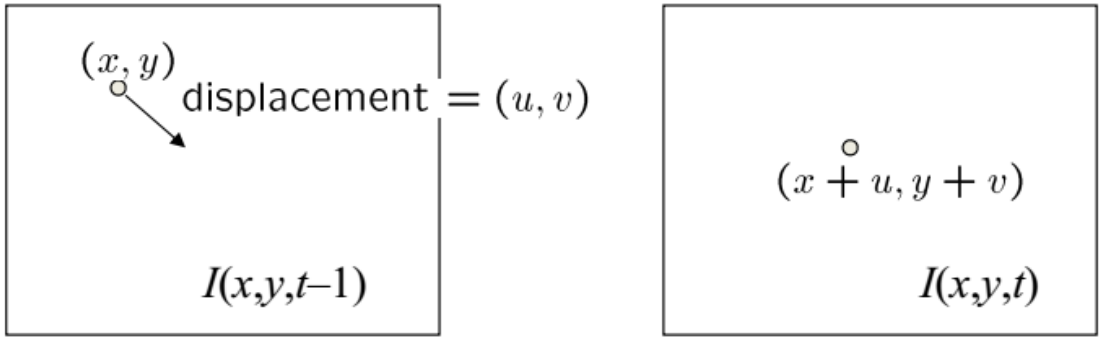
$$I(x, y, t-1) = I(x + u(x, y), y + v(x, y), t)$$

Linearizing the right side using Taylor expansion:

$$I(x, y, t-1) \approx I(x, y, t) + I_x\, u(x, y) + I_y\, v(x, y)$$

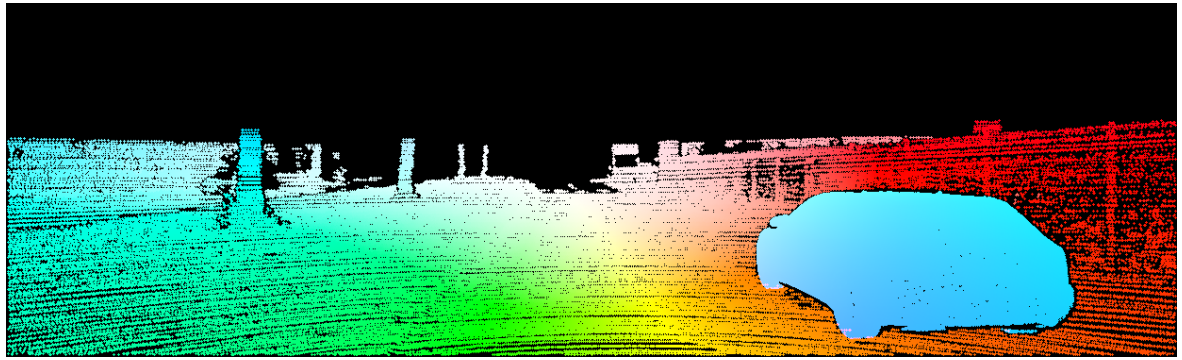
Hence, $I_x\, u + I_y\, v + I_t \approx 0$

One equation, two unknowns $\nabla I \cdot (u, v) + I_t = 0$



2. CNN models

- State-of-the-art:
  - Optical flow estimation



Method	Description	Performance
LDOF 2011 [5]	Optical Flow estimation	18.19 AEE KITTI15
DeepFlow 2013 [65]	Optical Flow estimation	10.63 AEE KITTI15
EpicFlow 2015 [53]	Optical Flow estimation	9.27 AEE KITTI15
FlowNet2 2016 [26]	Optical Flow estimation	8.94 AEE KITTI15

Endpoint Error (EE) is defined as the scalar length of different vectors between ||Vest – Vgt|| estimated optical flow vector Vest and groundtruth optical flow vector Vgt.
Average Endpoint Error (AEE) is the average of EE for all optical flow vector.

2. CNN models

- Public benchmarks & datasets

Datasets	Detection	Tracking	Segmentation	Optical Flow	Action recognition
KITTI 2012, 2015 [18, 45]	✓	✓	✓	✓	
Cityscapes 2016 [12]			✓		
CamVid 2008 [4]			✓		
DAVIS 2018 [7]			✓		
MOT Challenge 2016 [46]		✓			
PASCAL VOC 2012 [16]	✓		✓	✓	✓
COCO 2014, 2016, 2018[37]	✓		✓	✓	✓
VOT Challenge 2016 [31]		✓			

2. CNN models

- Public benchmarks & datasets

Datasets	Detection	Tracking	Segmentation	Optical Flow	Action recognition
MPI Sintel 2012 [6]				✓	
Flying chairs 2015 [15]				✓	
Scene Flow 2016 [44]				✓	
Middlebury 2015 [54]				✓	
HD1K 2016 [30]				✓	
ScanNet 2017 [13]			✓		
WildDash 2017 [69]			✓		
Hollywood2 2009 [43]					✓
UCF101 2012 [57]					✓
HMDB51 2011 [33]					✓

2. CNN models

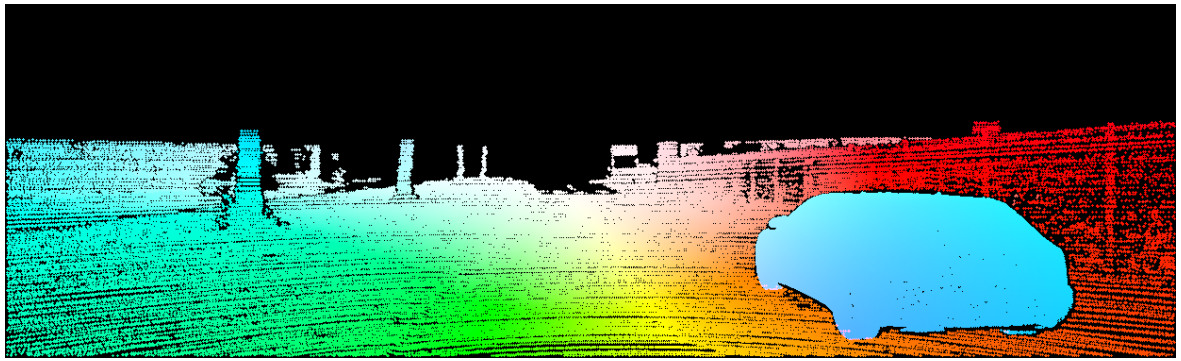
- Datasets illustration

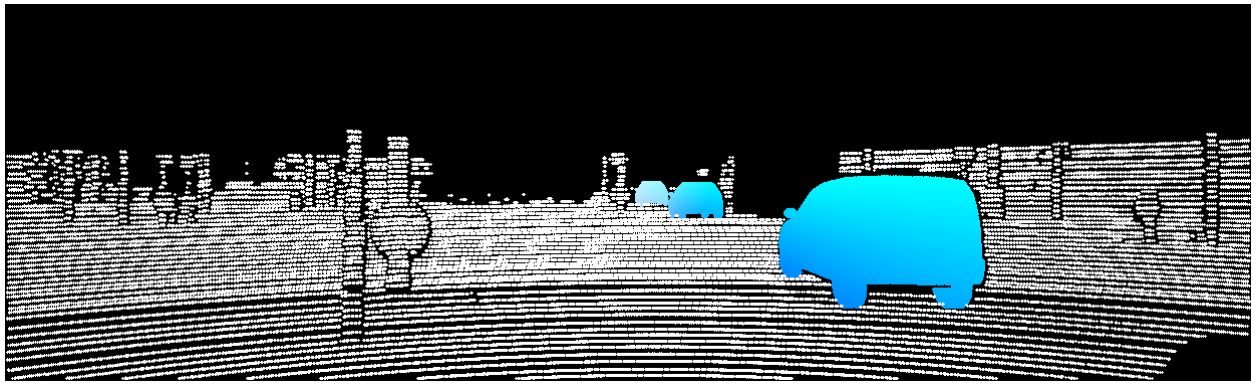
MOT Challenges

Cityscape dataset

2. CNN models

- Datasets illustration





KITTI DATASET

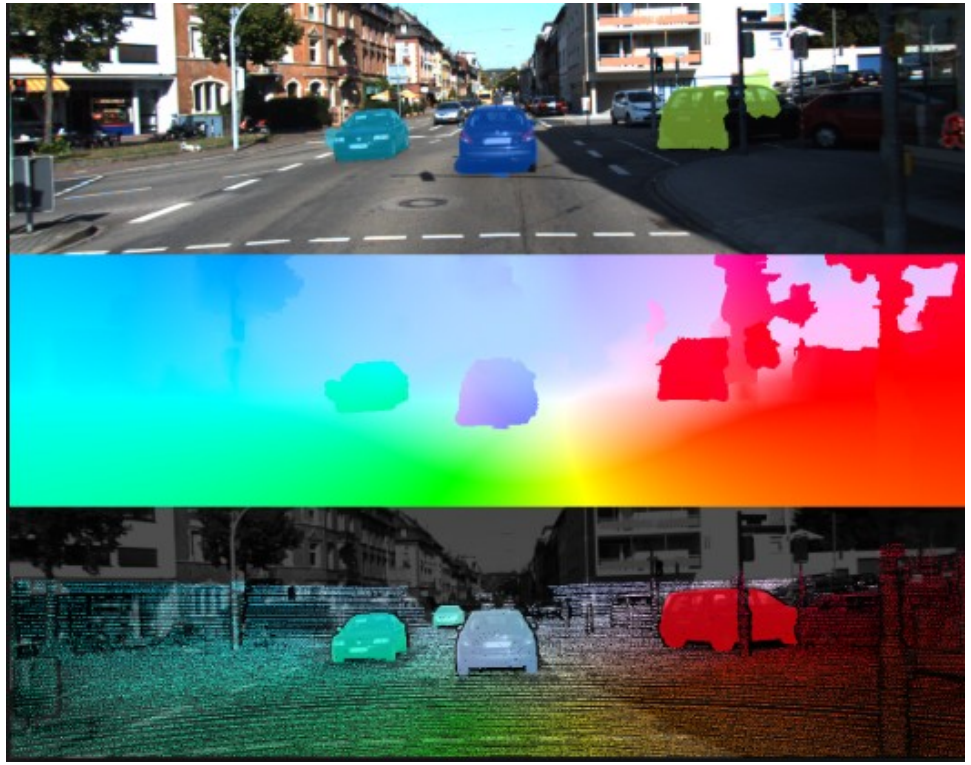
2. CNN models

- Datasets illustration

DAVID dataset

2. CNN models

- Datasets illustration

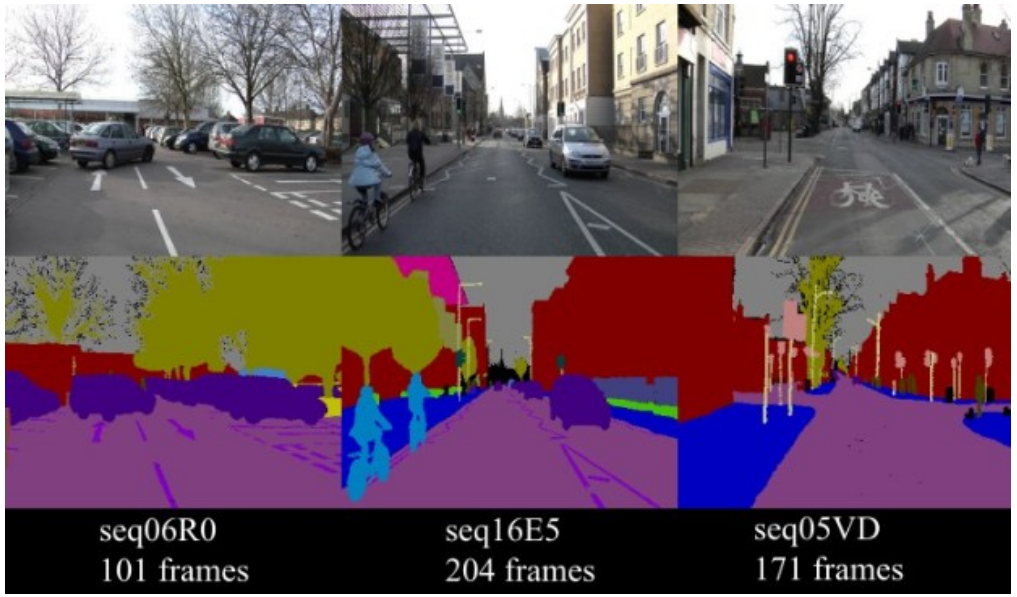


Scene Flow 2016 dataset

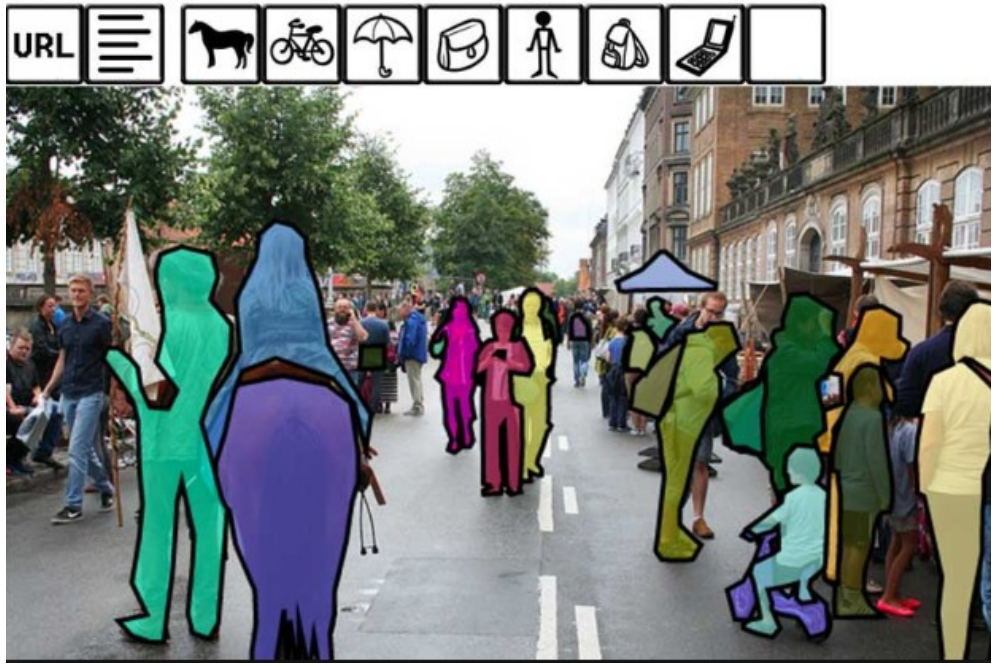
WildDash dataset

2. CNN models

- Datasets illustration



CamVid dataset



COCOdataset

Outline

1. Introduction
2. State of the art of CNN models
3. Instance segmentation with Mask RCNN
4. Improved tracking by predicting future segmentation
5. Future works
6. Conclusions

3. Instance segmentation Mask RCNN

Why do we need Mask RCNN ?

- Qualitative evaluation of some previous Deep Learning methods: AlexNet, VGG, ResNet; Faster RCNN, SegNet

Faster RCNN

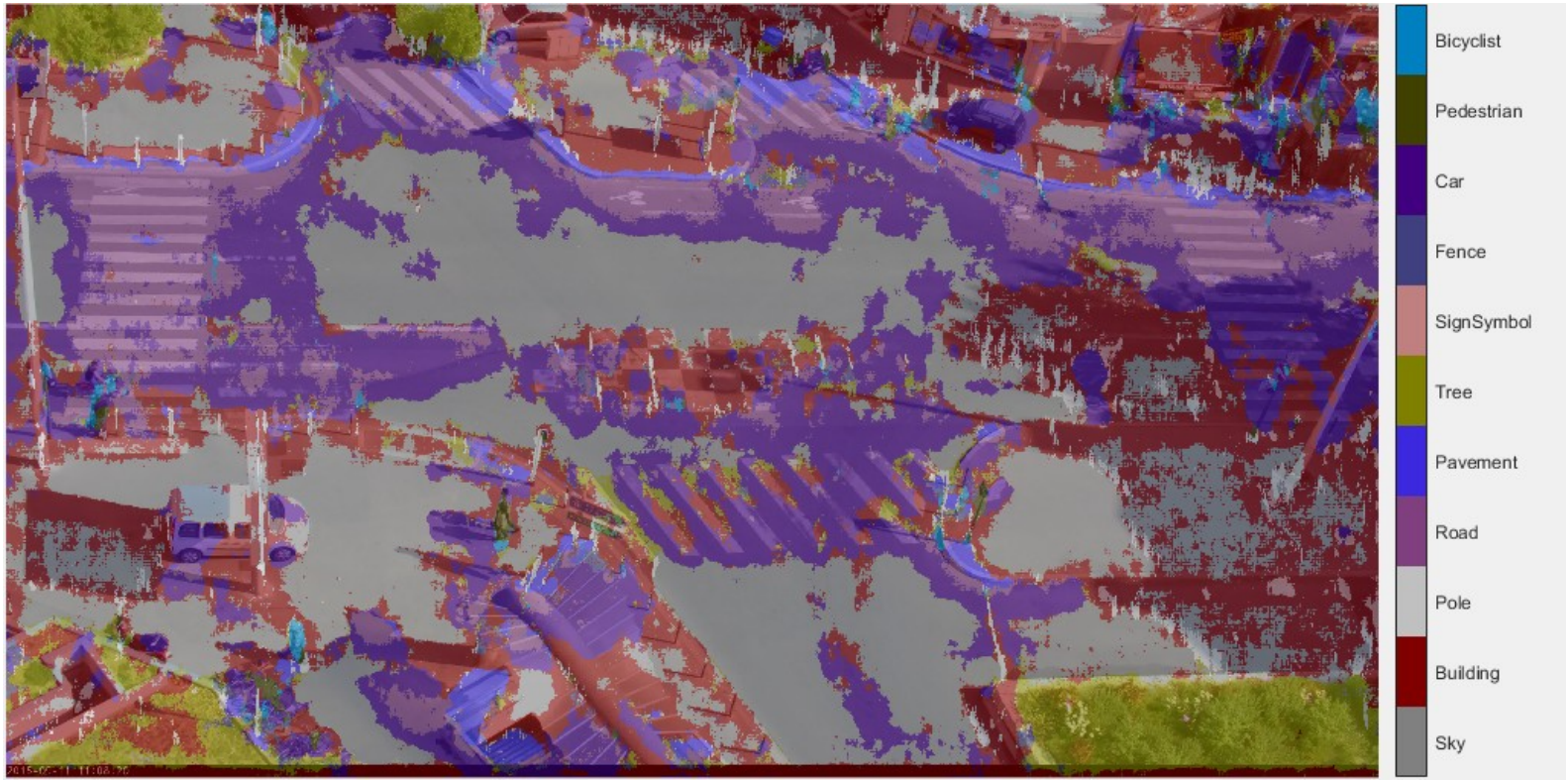
3. My works

- **Practical results:**

Faster RCNN

3. Instance segmentation Mask RCNN

- **Practical results:**



SegNet - Badrinarayanan 2015

3. Instance segmentation Mask RCNN

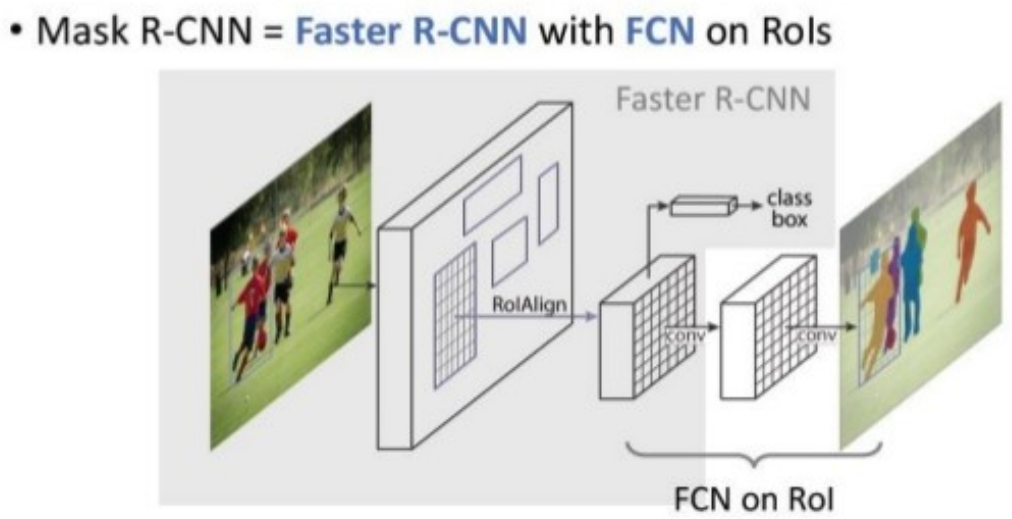
Mask RCNN

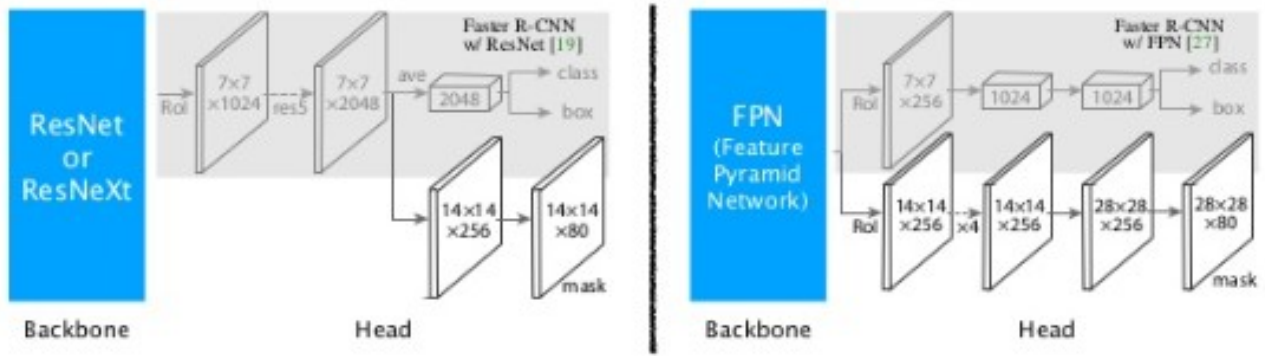
- Kaiming He et al. [CVPR 2017]
- Faster R-CNN: Detection ➔ Draw bounding box
- FCN: Segmentation ➔ Mask in bounding box

- Loss function

$$L = L_{cls} + L_{box} + L_{mask}$$

- ResNet backbone architecture





3. Instance segmentation Mask RCNN

- **Practical results:**

ORIO dataset

Static camera

Mask R-CNN – Kaiming He 2017

3. Instance segmentation Mask RCNN

Practical results:

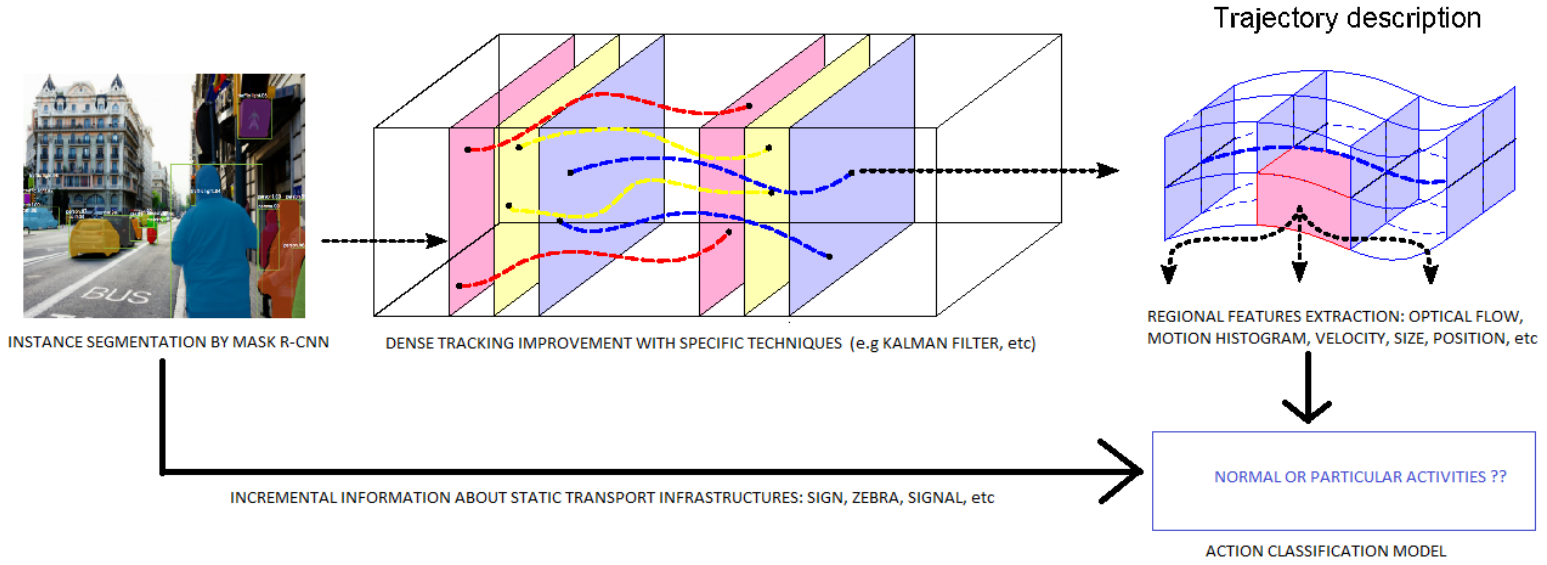
ORIO dataset

Moving camera

Mask R-CNN – Kaiming He 2017

3. Instance segmentation Mask RCNN

- General pipeline:
  - Using Mask RCNN to densely track road users & static transports infrastructure
  - Extract regional features to represent interactions, actions
  - Actions/Interactions classification



3. Instance segmentation Mask RCNN

- Drawbacks:
  - Discontinuity trajectories in tracking object with Mask RCNN

Outline

1. Introduction
2. State of the art of CNN models
3. Instance segmentation with Mask RCNN
4. Improved tracking by predicting future segmentation
5. Future works
6. Conclusions

4. Improve Tracking

- Problem:
  - Discontinuity trajectories in tracking object with Mask RCNN
- Why do we need to address this problem ?
  - Importance of dense and smooth trajectories in feature extraction for action recognition
  - Improve tracking technique in chaotic and ovelap scenario
  - Raise a new challenge: tracking with input information from some given frames
  - Link to generate problem: trajectories generating in the case that we can not access to future frames
  - Improve segmentation and detection with temporal information

4. Improve Tracking

Proposed solutions:

- Idea: generate the new segments in new frames based on the results of correct frames
- 4 approaches:
  - Optical flow
  - Deep neural network
  - Shape deformation
  - Time series processing

4. Improve Tracking

❑Optical Flow

➢**Optical flow**: Apparent motion of brightness patterns in the image. **Motion field:** the projection of the 3D scene motion into the image.

➔ ***Ideally, Optical Flow = Motion Field***

Brightness Constancy Equation:

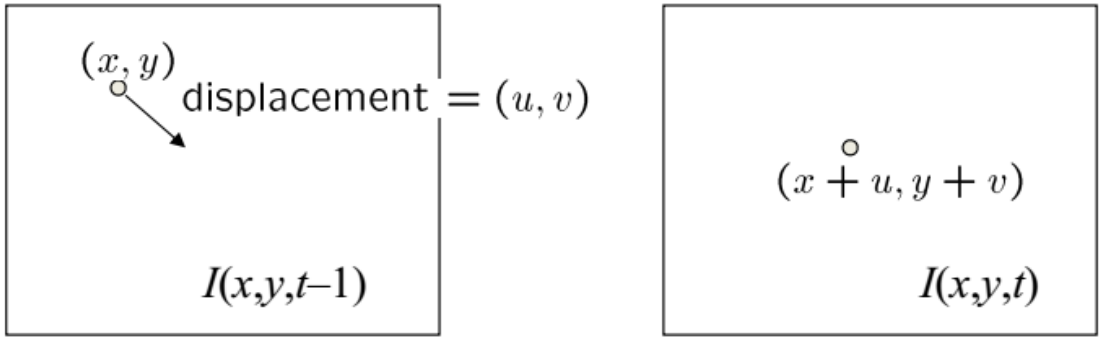
$$I(x, y, t-1) = I(x + u(x, y), y + v(x, y), t)$$

Linearizing the right side using Taylor expansion:

$$I(x, y, t-1) \approx I(x, y, t) + I_x\, u(x, y) + I_y\, v(x, y)$$

Hence, $I_x\, u + I_y\, v + I_t \approx 0$

One equation, two unknowns $\nabla I \cdot (u, v) + I_t = 0$



4. Improve Tracking

- **Optical flow approach:**
  - Extract optical flow (OF) vectors from given frames
  - Translate segments by new OF vectors: backward, forward, combining
  - Warp translate: Each pixel of the mask is translated with each OF vectors
  - Shift translate: All pixels of the mask is translated with average OF vectors

❖ Advantage: Simple method, take advantage from previous OF extraction

❖ Inconvenience: Sensitive with camera motion, non-rigid objects turning and illumination changing

❖ Paper: Predicting future instance segmentaions by forecasting convolutional features (**Pauline Luc et al.** ECCV 2018)

4. Improve Tracking

- **Experiment setup:**
  - Run Mask RCNN for DAVIS dataset
  - Randomly discard some frames to make the missing-frame situation
  - Using Optical Flow to generate new instance segmentations (LDOF, Full Flow, PwC-Net)
  - Comparing generated segments with Mask RCNN segments: mIOU

4. Improve Tracking

- **Results on small part of DAVIS dataset**

Methods	Warp (mIOU %)	Shift (mIOU %)
Backward	84.42	89.22
Forward	84.18	88.65
Combine-Forward	**84.70**	**89.56**
Combine-Backward	84.47	89.21

4. Improve Tracking

- Qualitative results

T-1

T

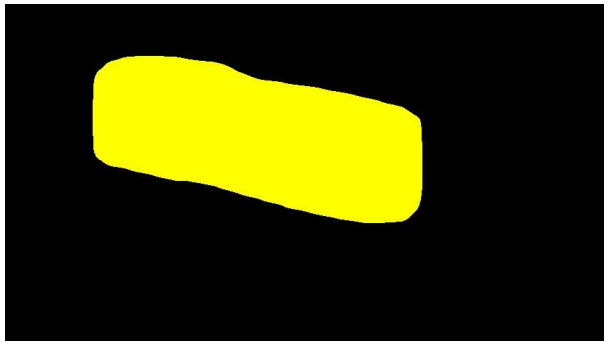
T+1

OF (t-1,t)

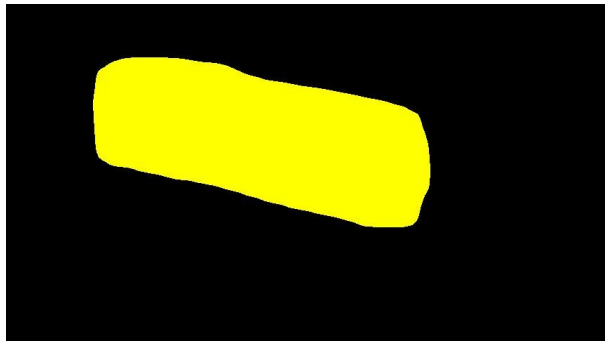
OF (t,t+1)

4. Improve Tracking

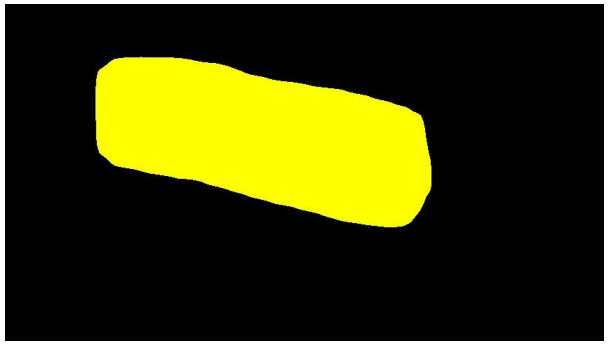
- Qualitative results



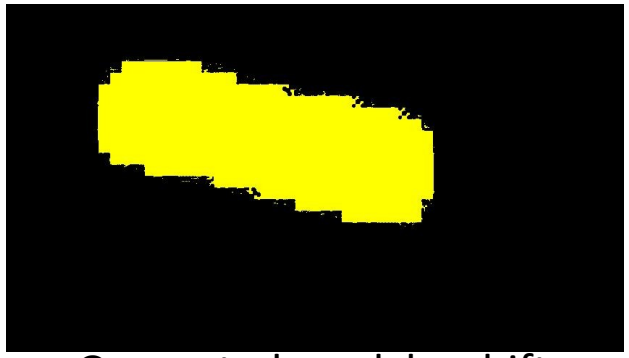
Mask t-1



Mask t



Mask t+1



Generated mask by shift

Generated mask by warp

4. Improve Tracking

- **Problem expansion:**
  - **Ideal case:**

$$S_t = w_1 T(S_{t-1}) + w_2 T(S_{t+1})$$

➔ Learn the scalar weight w1, w2 with traditional learning methods

  - **Deal with general problem (camera motion, non-rigid objects turning)**

$$S_t = \sum_{i,j} (w_i(T,\theta) S_{t-i} + w_j(T,\theta) S_{t+j})$$

5. Future works

❖CNNs Approach:

- Create a neural network (GAN model, encode-decode model), given input as correct frames, achieve output as new segments of new frames
- Prepare suitable training and test set
- Learn the model and evaluate the qualitative performance in our scenario

❖ Advantages: Take advantages from recently stronger and flexible CNNs architecture

❖ Inconvenience: Prepare dataset; complexity methods, time consuming

❖Paper: Predicting future instance segmentaions by forecasting convolutional features.ECCV 2018; GANs papers; One-shot video object segmentation,CVPR 2017

5. Future works

- Shape deformation approach:
  - Create geometrical model of correct frames
  - Solve the matrices of translation, rotation, scale changing

- Advantages: Take advantages from simple computer graphic methods
- Inconvenience: Change in some parties of the object
- Paper: As-rigid-as-possible Shape Manipulation

5. Future works

- Time series approach:
  - Kalman filter
  - Long short term memories

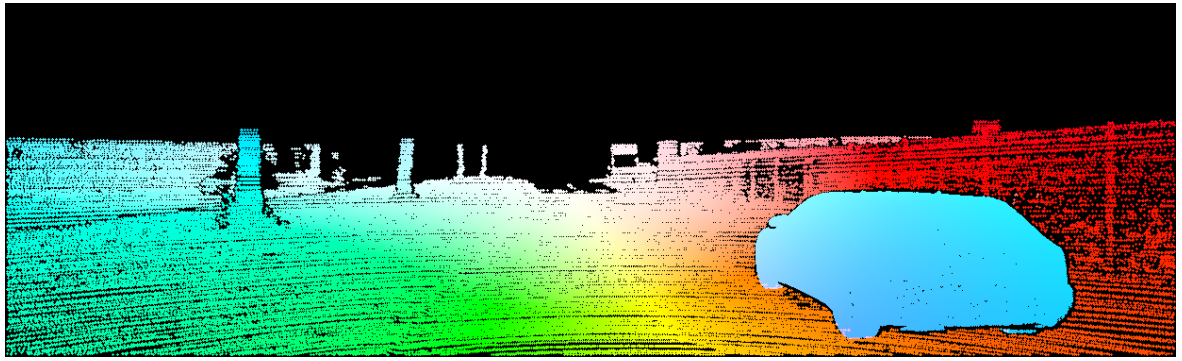
5. Conclusion

- Our challenging problem is addressed as "**visual action understanding**" and approached by **Deep Learning** methods.
- Mask R-CNN is a promising candidate for constructing solutions → propose a joint segmentation and classification solution.
- Mask R-CNN can be improved by combining it with other specific techniques (Optical flow, etc) for dense tracking.

THANK YOU
FOR
YOUR ATTENTION

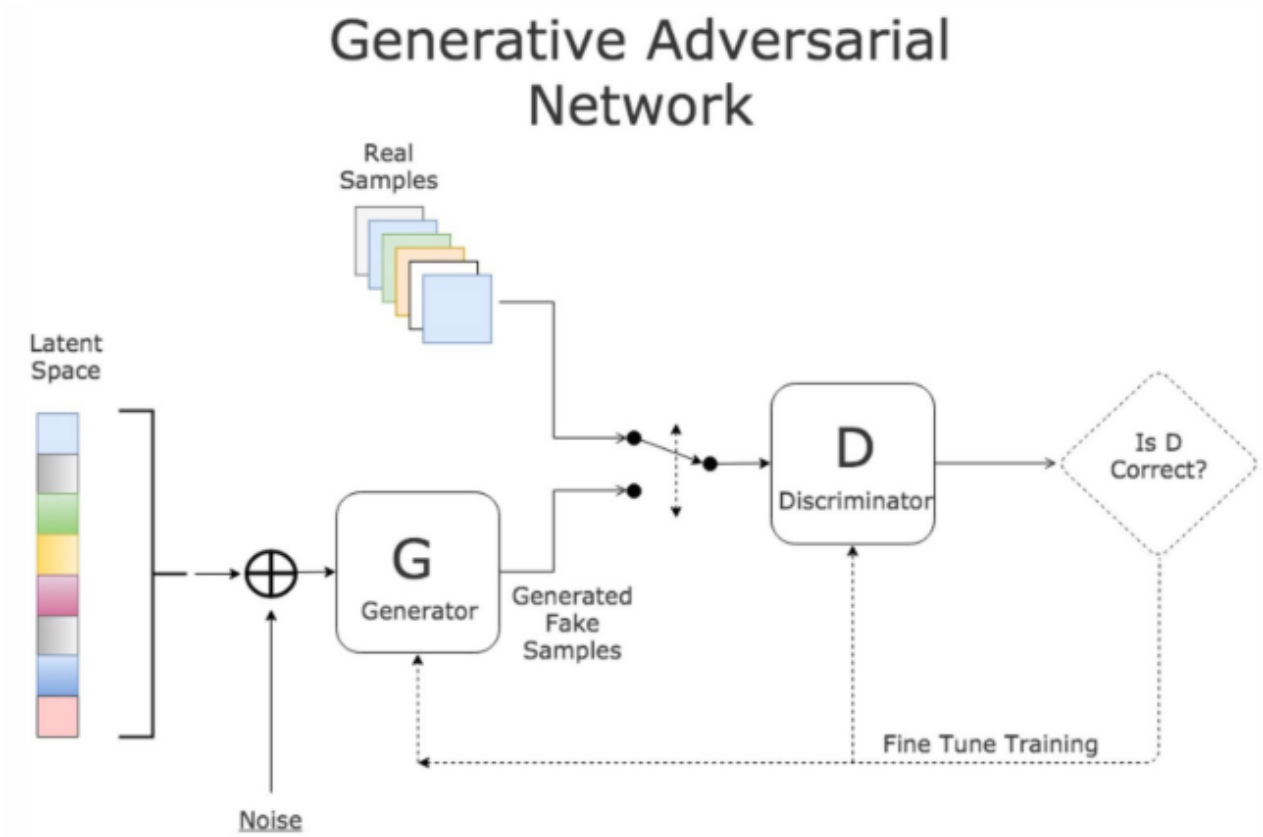
DICUSSION

- Future plan:
  - Solving general problem Optical flow
  - New generating model
  - More tracking technique
  - Interaction modeling



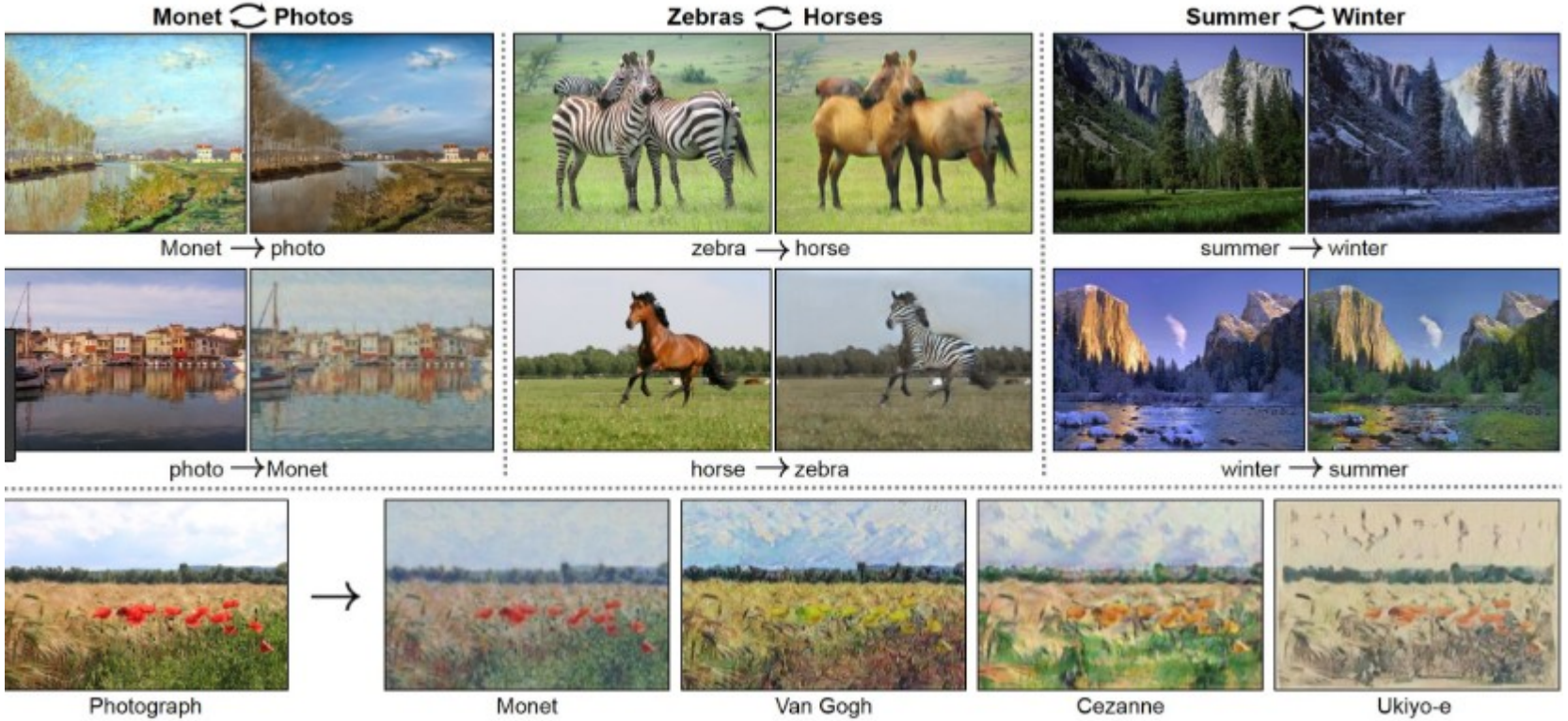
KITTI DATASET

DICUSSION



https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html

DICUSSION



CycleGAN – JY Zhu et al. ICCV 2017

Training feed-forward neural network

- Non-convex optimization problem in general (or at least in useful cases)
  - Typically number of weights is (very) large (millions in vision applications)
  - Seems that many different local minima exist with similar quality

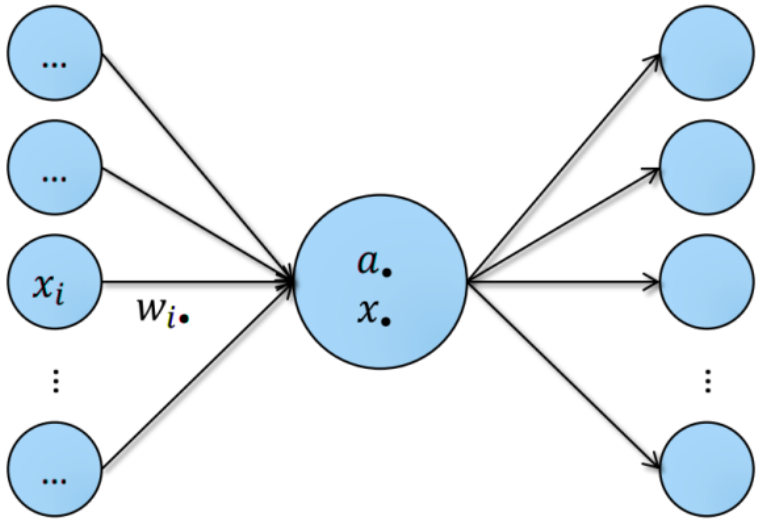
$$\frac{1}{N}\sum_{i=1}^{N} L(f(x_i), y_i; W) + \lambda \Omega(W)$$

- Regularization
  - L2 regularization: sum of squares of weights
  - “Drop-out”: deactivate random subset of weights in each iteration
    - Similar to using many networks with less weights (shared among them)

- Training using simple gradient descend techniques
  - Stochastic gradient descend for large datasets (large N)
  - Estimate gradient of loss terms by averaging over a relatively small number of samples

Training the network: forward propagation

- Forward propagation from input nodes to output nodes
  - Accumulate inputs into weighted sum
  - Apply scalar non-linear activation function f
- Use Pre(j) to denote all nodes feeding into j



$$a_j = \sum_{i \in Pre(j)} w_{ij} x_i$$

$$x_j = f(a_j)$$

Training the network: backward propagation

- Input aggregation and activation

$$a_j = \sum_{i \in Pre(j)} w_{ij} x_i$$

$$x_j = f(a_j)$$

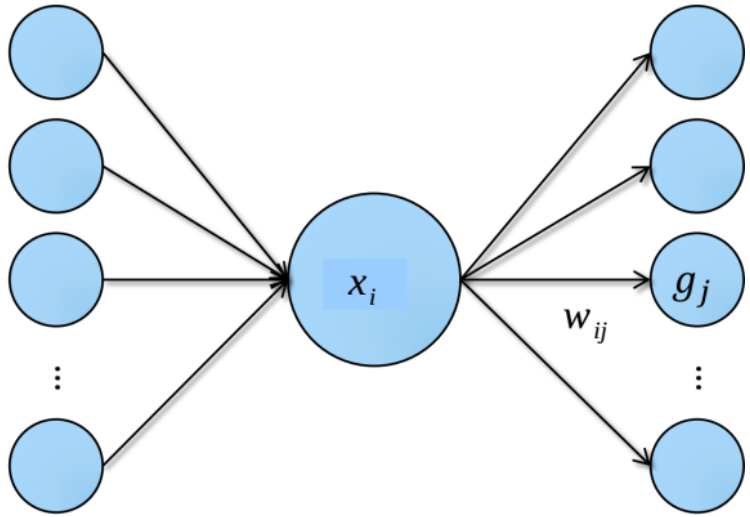
- Partial derivative of loss w.r.t. input

$$g_j = \frac{\partial L}{\partial a_j}$$

- Partial derivative w.r.t. learnable weights

$$\frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial a_j} \frac{\partial a_j}{\partial w_{ij}} = g_j x_i$$

- Gradient of weights between two layers given by outer-product of x and g



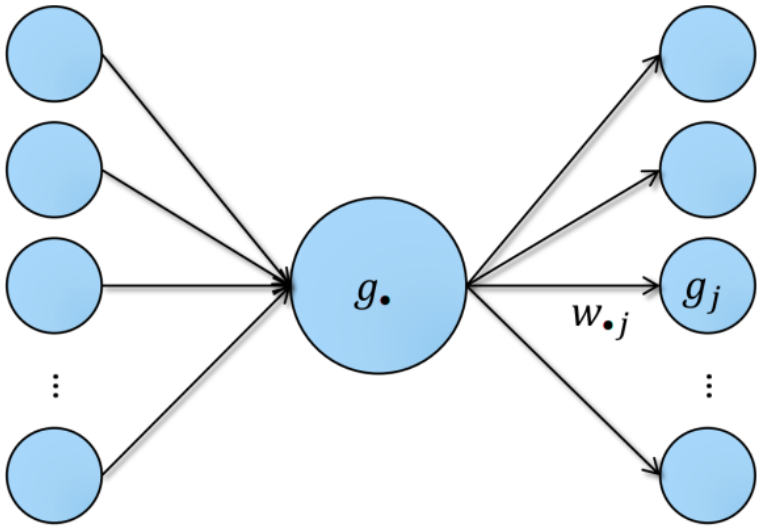
Training the network: backward propagation

- Backward propagation of loss gradient from output nodes to input nodes
  - Application of chainrule of derivatives
- Accumulate gradients from downstream nodes
  - Post(i) denotes all nodes that i feeds into
  - Weights propagate gradient back
- Multiply with derivative of local activation

$$a_j = \sum_{i \in Pre(j)} w_{ij} x_i$$

$$x_j = f(a_j)$$

$$g_i = \frac{\partial L}{\partial a_i}$$



$$\frac{\partial L}{\partial x_i} = \sum_{j \in Post(i)} \frac{\partial L}{\partial a_j} \frac{\partial a_j}{\partial x_i}$$
$$= \sum_{j \in Post(i)} g_j w_{ij}$$

$$g_i = \frac{\partial x_i}{\partial a_i} \frac{\partial L}{\partial x_i}$$
$$= f'(a_i) \sum_{j \in Post(i)} w_{ij} g_j$$

Training the network: forward and backward propagation

- Special case for Rectified Linear Unit (ReLU) activations

$$f(a)=max(0,a)$$

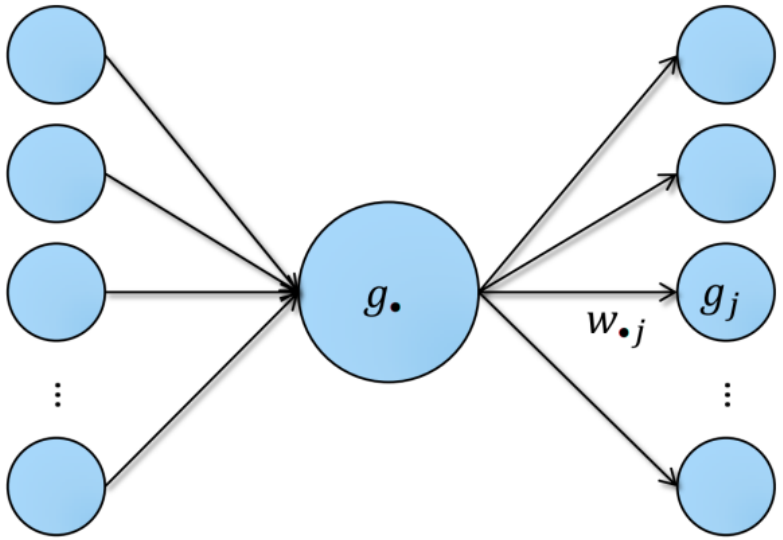
- Sub-gradient is step function

$$f'(a)=\begin{cases}0 & \text{if } a\leq 0\\ 1 & \text{otherwise}\end{cases}$$

- Sum gradients from downstream nodes

$$g_i=\begin{cases}0 & \text{if } a_i\leq 0\\ \sum_{j\in Post(i)} w_{ij}g_j & \text{otherwise}\end{cases}$$

  - Set to zero if in ReLU zero-regime
  - Compute sum only for active units



- Note how gradient on incoming weights is "killed" by inactive units
  - Generates tendency for those units to remain inactive

$$\frac{\partial L}{\partial w_{ij}}=\frac{\partial L}{\partial a_j}\frac{\partial a_j}{\partial w_{ij}}=g_j x_i$$