# SLIDING CONVOLUTION WINDOW

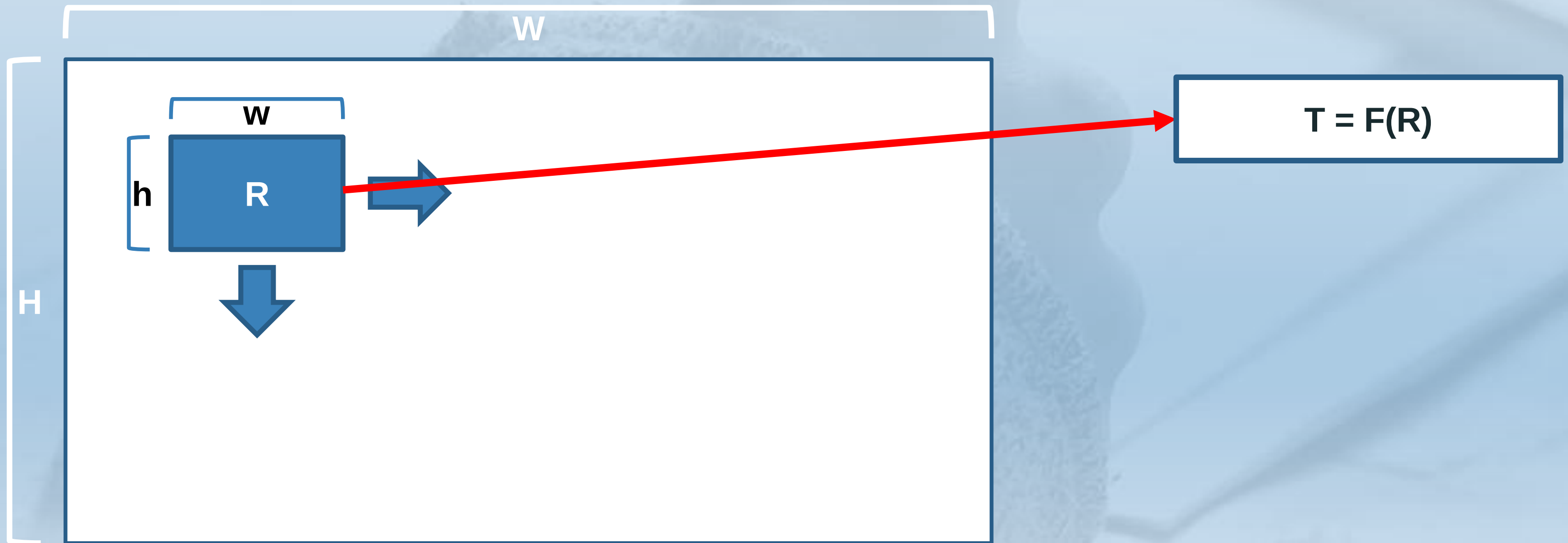
WENHAO HE
Jan 4th 2016

# Outline

- Sliding Window Model
- Fully Convolution Network (FCN)
- Densebox
- Applications in scene text detection
- Trend and future
- Brief introduction of BD-IDL

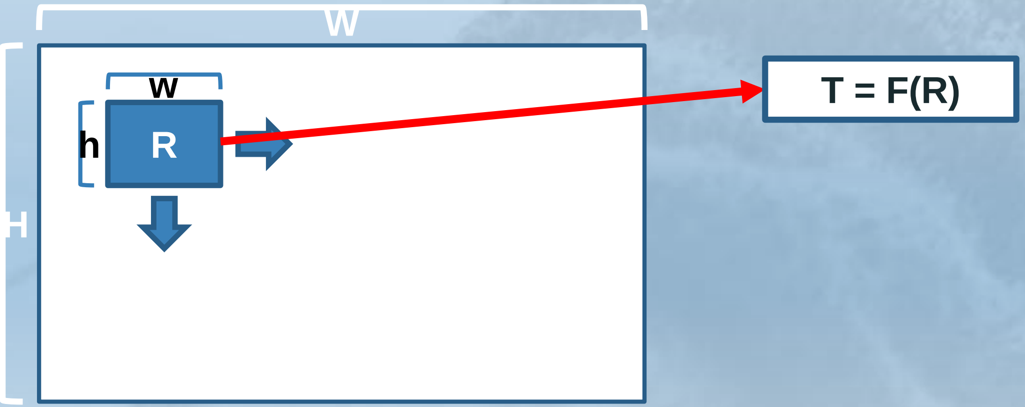
# Sliding Window Model

- Given a map of size H×W and a window of size h×w
- Slide the window by stride *s* in both x and y directions
- The intersection region *R* will output a struct *T* by a function F. *T* = F(*R* )



# Sliding Window Model

- If *T* is a real value, we will get sliding window method in conventional object (text) detection and *T* refers to the object (text) region confidence



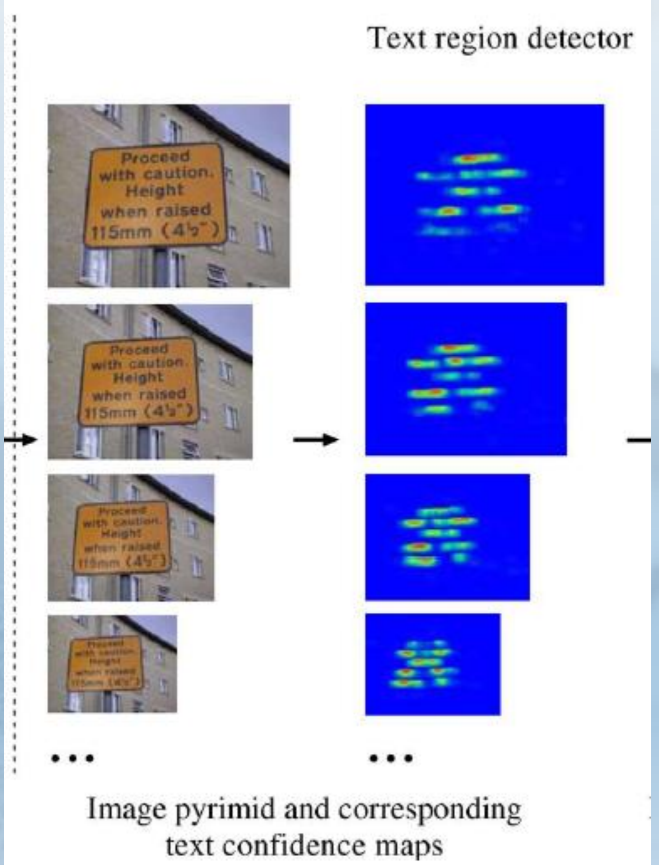
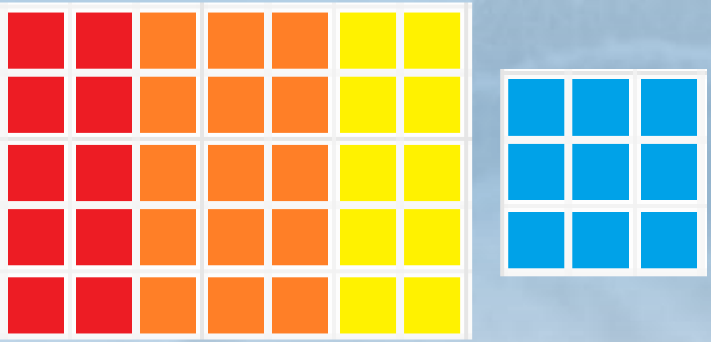
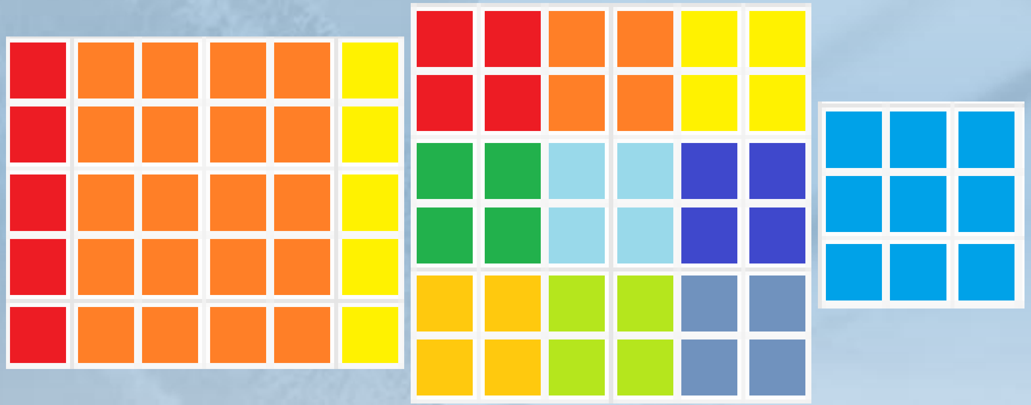


Image pyrimid and corresponding text confidence maps

# Convolution Network

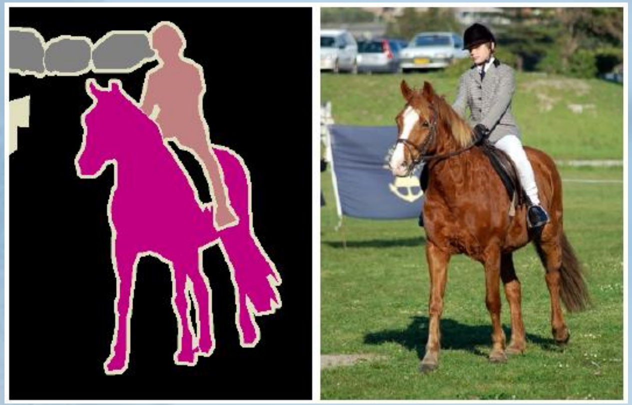
- If *T* is a real value and F is a convolution operation, we will get conventional CNN feature extractor
- If F is conv5×5, then h=w=5 (window size)
- If F is conv5×5, *s*=2→conv3×3, then h=w=5+2×2
- If F is conv5×5, *s*=1→pool2×2→conv3×3, then h=w=5+5×1
- If F is conv appended by MLP, we will get CNN





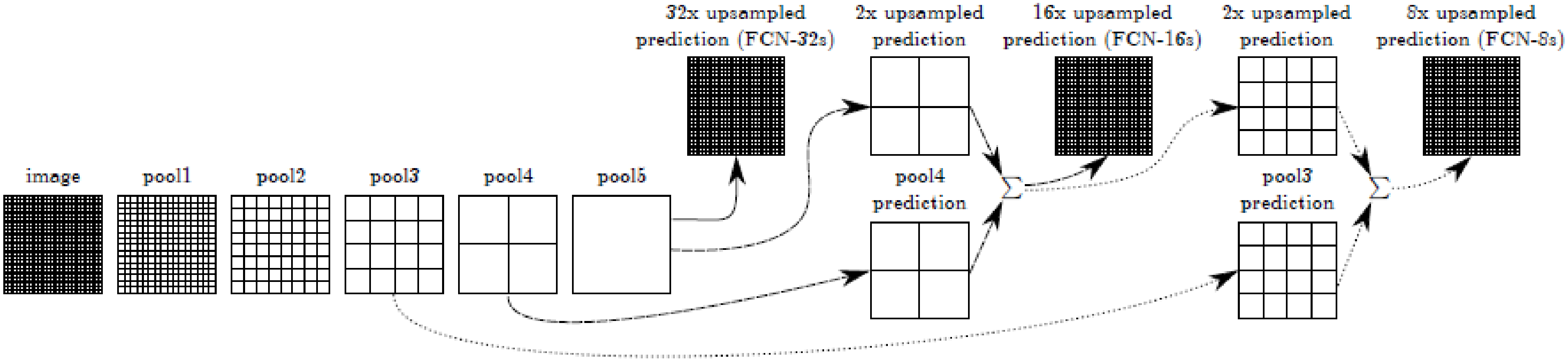
# Fully Convolution Network

- If $T$ is a 3-dim matrix, F is a convolution operation and stride $s$=+∞, we will get Fully Convolution Network
- FCN is proposed for segmentation task
- If the segmentation task deals with N categories, $T$ is a W×H×N matrix
- Each 1×1×N vector of $T$ refers to the probability of each class for a pixel
- We can also set $T$ to be W×H neglecting probability information



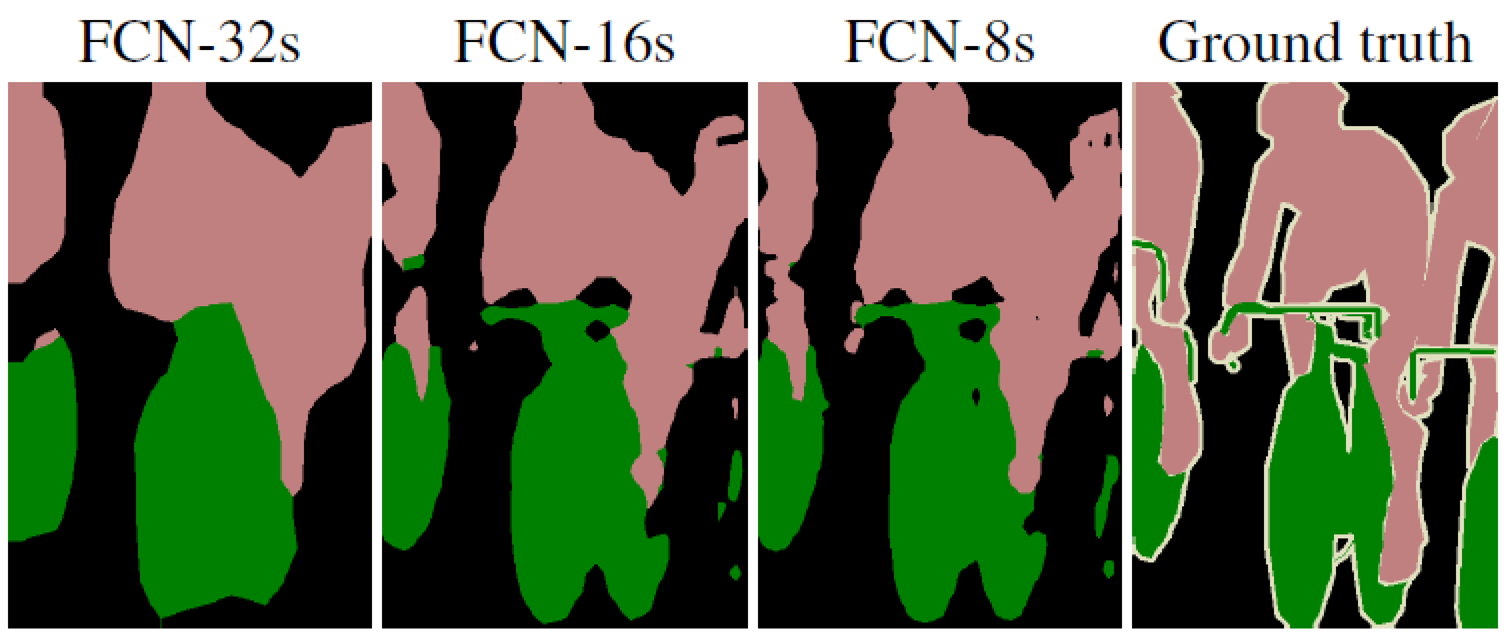
# Fully Convolution Network

- $\text{Pool}_N$ is deconved to fuse with $\text{Pool}_{N-1}$
- Larger N contains more category information (overall level) but loses details (pixel level) and vice versa



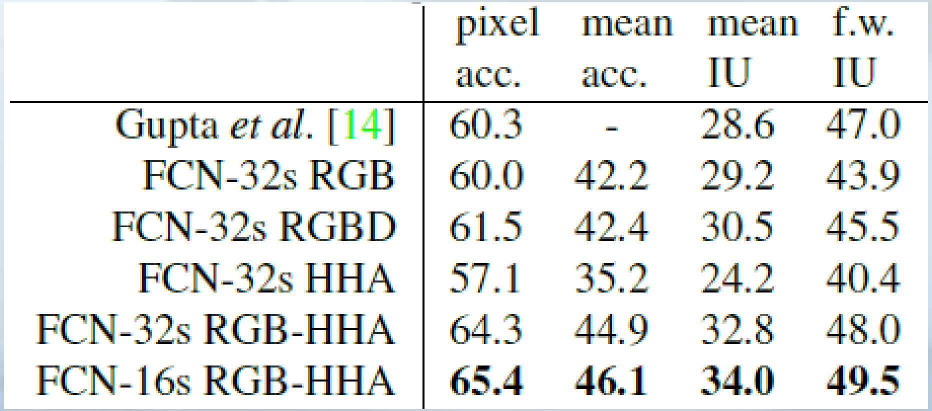
# Fully Convolution Network

- $\text{Pool}_N$ is deconved to fuse with $\text{Pool}_{N-1}$
- Larger N contains more category information (overall level) but loses details (pixel level) and vice versa



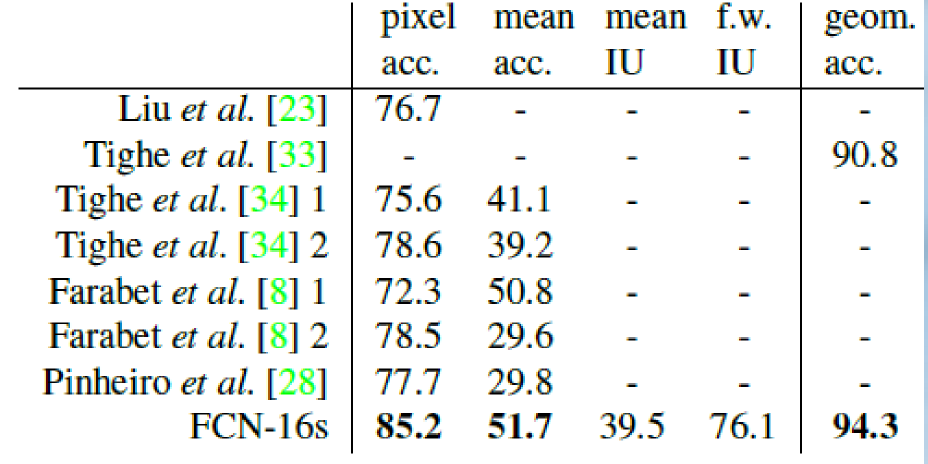
# Fully Convolution Network

- NYUDv2
  - □ Containing depth information



| | pixel acc. | mean acc. | mean IU | f.w. IU |
|---|---|---|---|---|
| Gupta *et al.* [14] | 60.3 | - | 28.6 | 47.0 |
| FCN-32s RGB | 60.0 | 42.2 | 29.2 | 43.9 |
| FCN-32s RGBD | 61.5 | 42.4 | 30.5 | 45.5 |
| FCN-32s HHA | 57.1 | 35.2 | 24.2 | 40.4 |
| FCN-32s RGB-HHA | 64.3 | 44.9 | 32.8 | 48.0 |
| FCN-16s RGB-HHA | **65.4** | **46.1** | **34.0** | **49.5** |

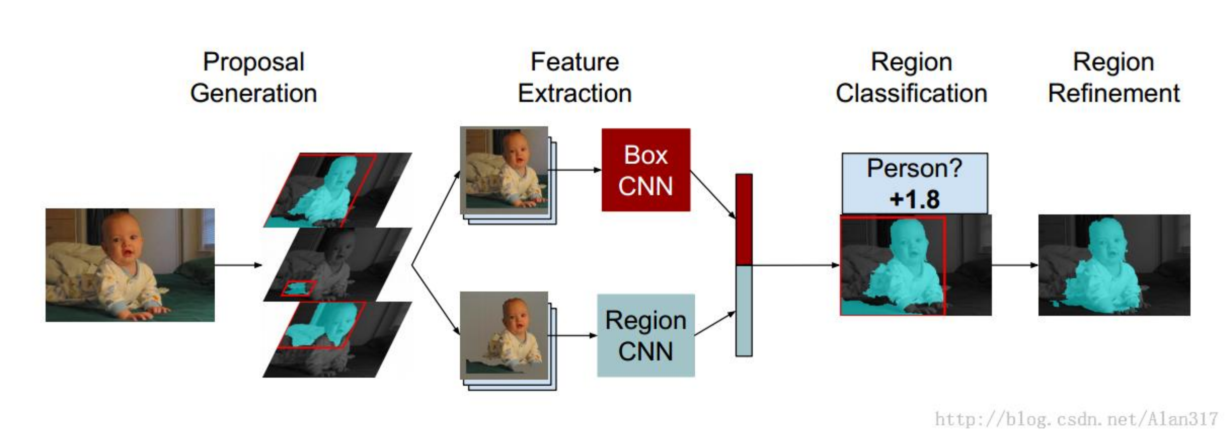
- Sift Flow

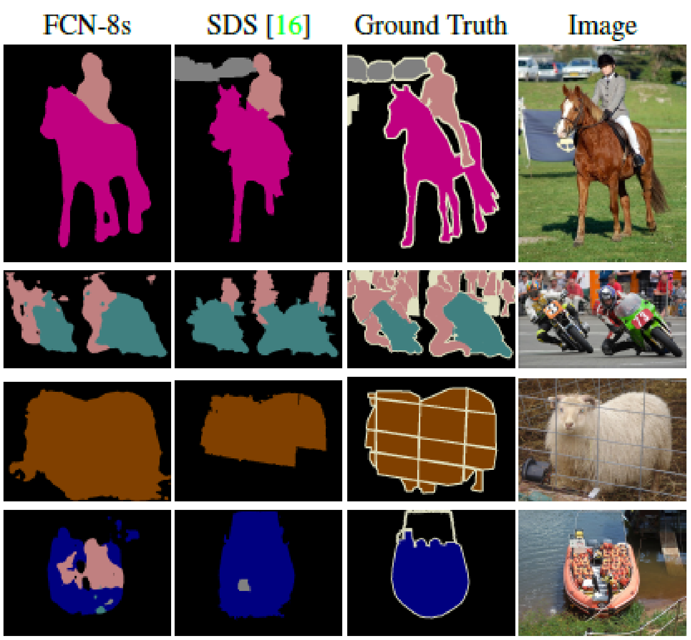


| | pixel acc. | mean acc. | mean IU | f.w. IU | geom. acc. |
|---|---|---|---|---|---|
| Liu *et al.* [23] | 76.7 | - | - | - | - |
| Tighe *et al.* [33] | - | - | - | - | 90.8 |
| Tighe *et al.* [34] 1 | 75.6 | 41.1 | - | - | - |
| Tighe *et al.* [34] 2 | 78.6 | 39.2 | - | - | - |
| Farabet *et al.* [8] 1 | 72.3 | 50.8 | - | - | - |
| Farabet *et al.* [8] 2 | 78.5 | 29.6 | - | - | - |
| Pinheiro *et al.* [28] | 77.7 | 29.8 | - | - | - |
| FCN-16s | **85.2** | **51.7** | 39.5 | 76.1 | **94.3** |

# Fully Convolution Network

- SDS (ECCV 2014)
  - □ Simultaneous Detection and Segmentation
  - □ Not an end to end method





# Densebox

- If *T* is a W×H×N struct*{conf, bbox}* for FCN structure, we will get Densebox
- Densebox was proposed for detection task, so N=1 and *T.conf* refers to the confidence map and *T.bbox* refers to the bounding boxes
- Actually Densebox differs from FCN in many details
  - Dimension of *T*
  - Label design
  - Training procedure
  - Testing procedure

# Details

■ Dimension of $T$

□ Suppose the input is a 640×480×3 color image

| Model | FCN | Densebox |
|---|---|---|
| Input size | 640×480×3 | 640×480×3 |
| Output size | 640×480×N | 160×120×struct |

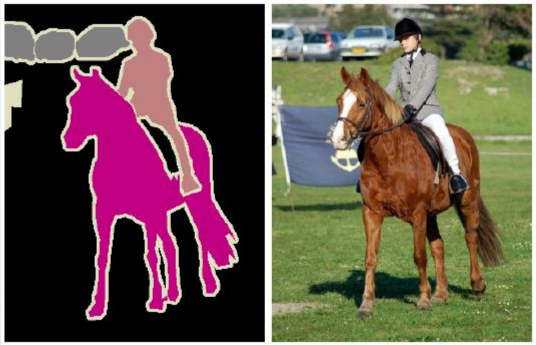
□ Detection task is required less detailed information

□ However, Densebox can output a 640×480 struct if we try to get both segmentation and detection information

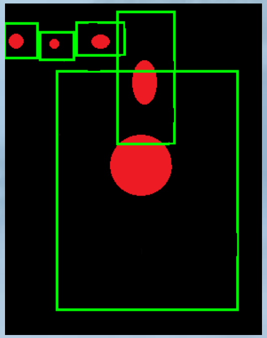
□ For human visual system, we do segmentation and detection simultaneous. Multi-task is a trend.

# Details

- ■ Label design
  - □ FCN

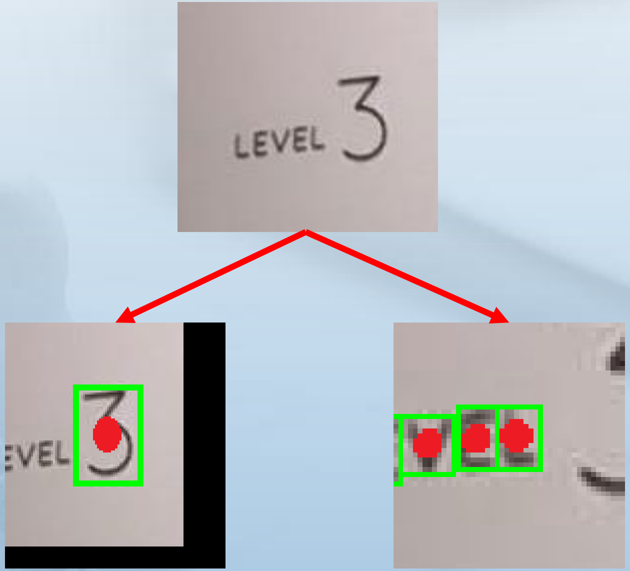


  - □ Densebox



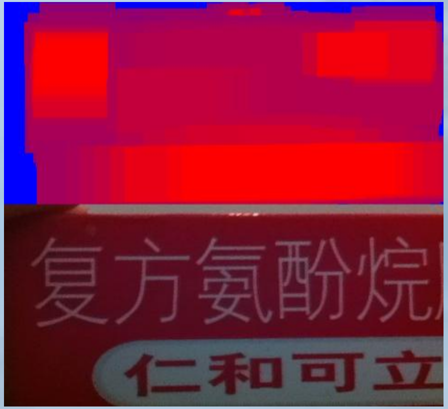
# Details

- Training procedure
  - □ FCN
    - Single task
    - Softmax loss for per pixel
  - □ Densebox
    - ROI is cropped out as a positive sample
    - Each bounding box size is normalized
    - Too big or small ones are negative
    - Multi-task: detection and regression
    - Detection: Hinge loss
    - Regression: Euclidean  loss



# Details

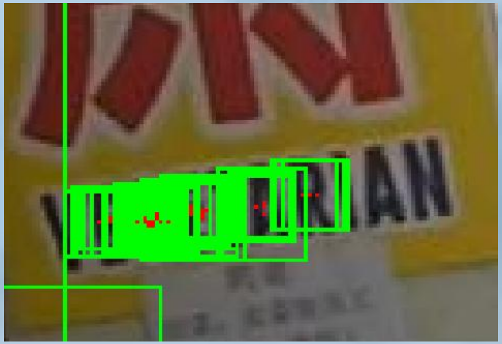
- Training procedure
  - Hard negative sampling
  - Rank the scores of negative regions by descend order
  - The high score negative regions are hard to classify
  - Ignore the backward gradient from low score negative region

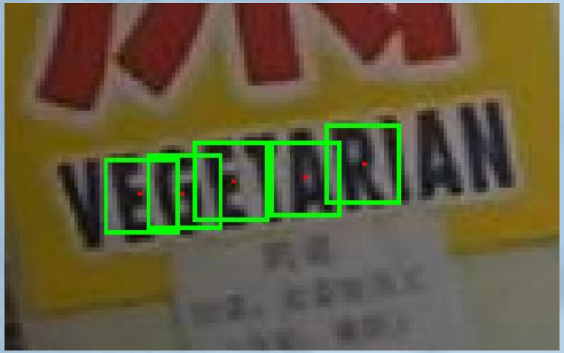


# Details

- Testing procedure
  - FCN
    - Directly output a W×H×N matrix

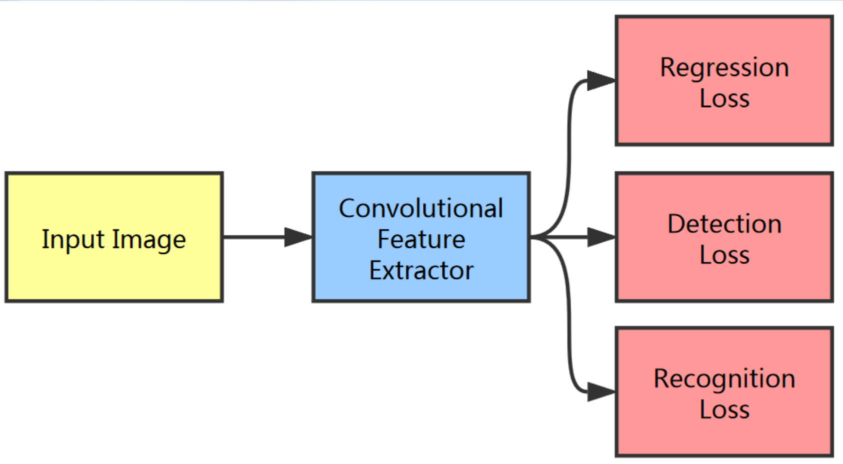
  - Densebox
    - Multi-scale
    - Output confidence and a bounding box for each pixel (Densebox)
    - NMS to mine the dense boxes





# Math Expression recognition

- ■ Adopt Densebox structure
- ■ Comparison with traditional model
- □ Auto cut torching part
- □ Using context → log or 10g
- □ Auto combine multi-parts → - to =

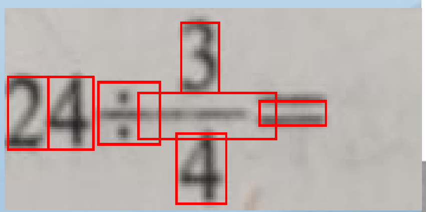
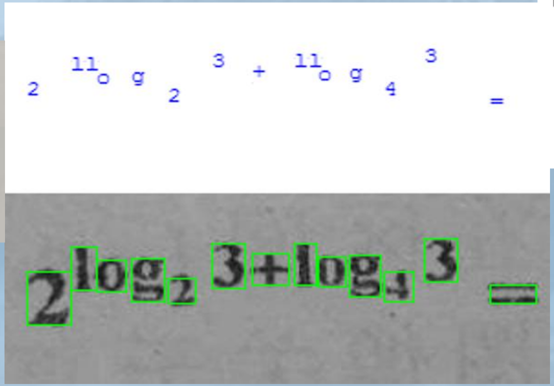

Input Image

Convolutional Feature Extractor

Regression Loss

Detection Loss

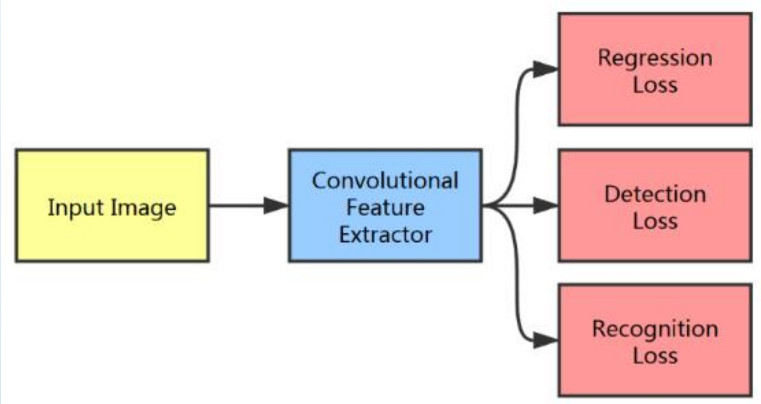
Recognition Loss

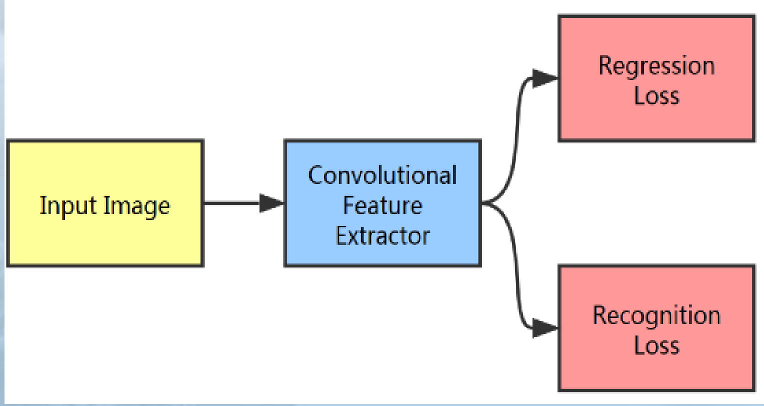



# Math Expression recognition

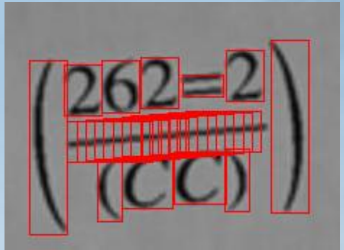
■ Details

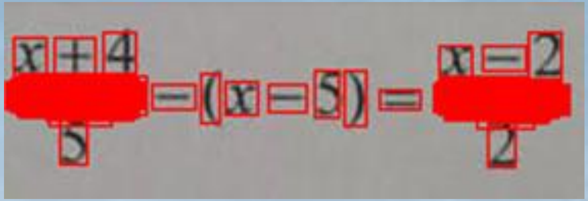
□ How many tasks should we use

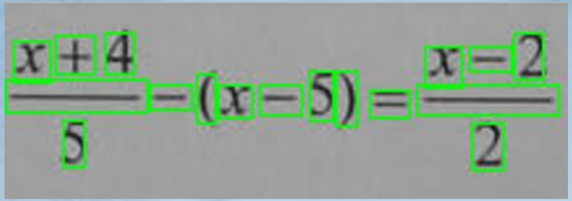




□ Fraction line (exaggerated Width/Height ratio)





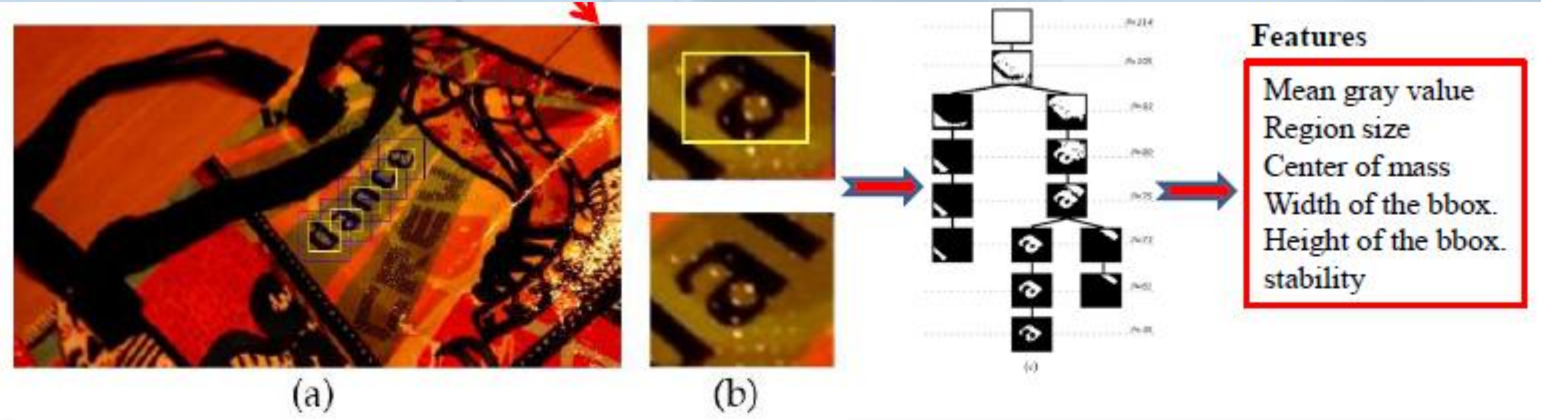


# Scene text (line) detection

- ■ Scene text detection is a very typical problem
  - □ A specific area for object detection and recognition
  - □ Scales and transformation of text vary much
  - □ Text line detection requires sequence learning (In my view, we need recognition result to refine text line)

# Scene text (line) detection

- Proposal based method
  - Classify MSER proposals lose too much context information
  - Recently, methods like fast-RCNN and faster-RCNN using context features are popular



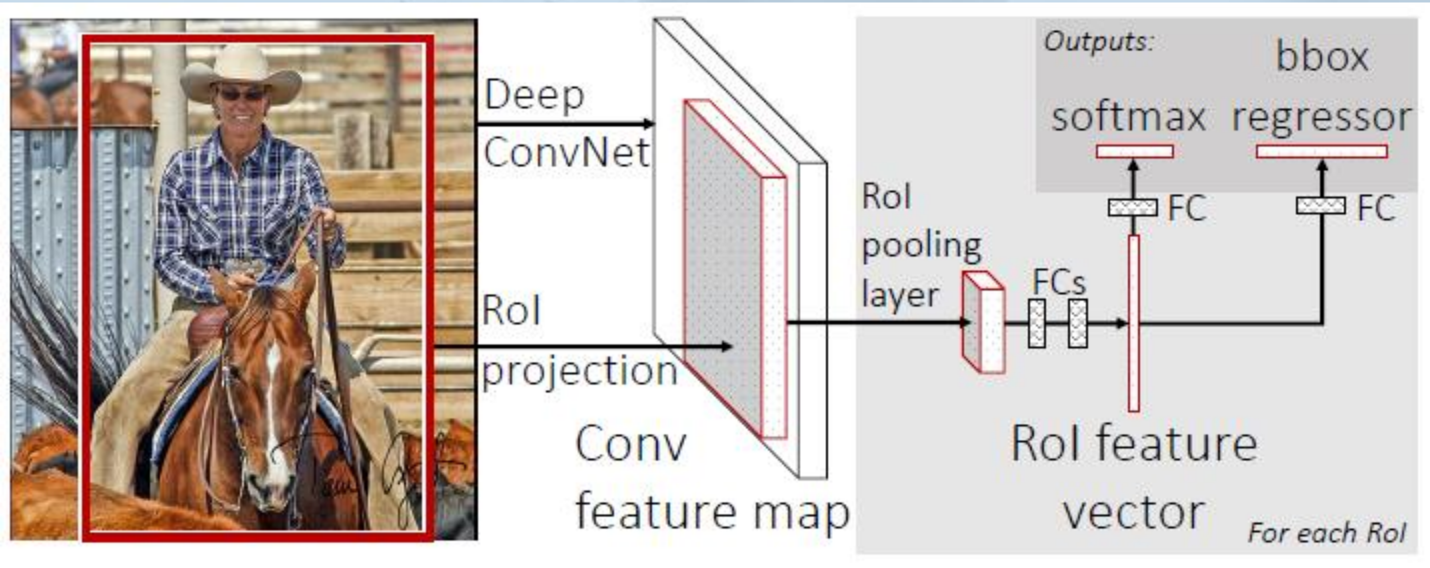
(a) (b)

**Features**

Mean gray value
Region size
Center of mass
Width of the bbox.
Height of the bbox.
stability

# Scene text (line) detection

- ■ Proposal based method
- □ Classify MSER proposals lose too much context information
- □ Recently, methods like fast-RCNN and faster-RCNN using context features are popular (CNN features of the whole image are extracted beforhand)



# Scene text (line) detection

- Sliding window method
  - Boosting methods
  - Densebox is a specific sliding window model by the our defination
  - Actually fast-RCNN and faster-RCNN methods can also be regarded as a special sliding window model for certain regions (MSER regions)

# Scene text (line) detection

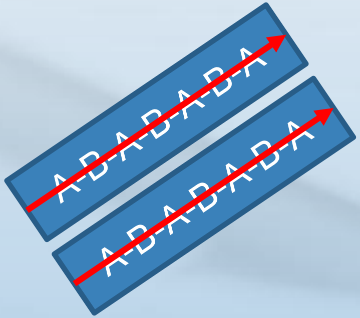
■ How many types of information should and could we get?

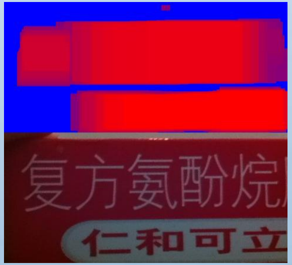
□ Single letter’s bounding box

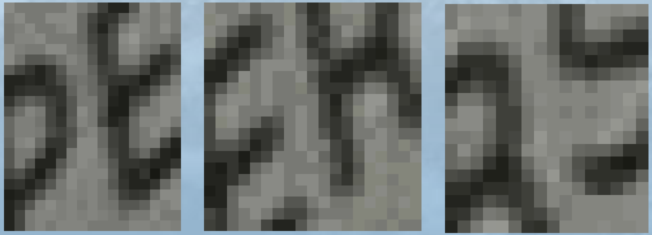
□ Single letter’s rotation

□ Single letter’s category

□ Single letter’s segmentation [Text attention CNN]





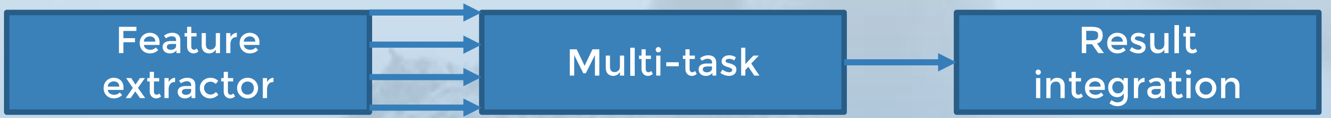


□ Pair part between letters : **Text Line**

▪ Text line is arranged in such format:
A-B-A-B-A-B-A (A: single word, B: pair part)

# Two schemes

- Pure detection
  - We also use category information for each letter as one of the multi-tasks, but pay less attention for the accuracy
  - We do not use recognition result to refine text line



Feature extractor

Multi-task

Result integration

- End to end system for both detection and recognition
  - Sequence learning should be taken into consideration
  - Refine single letter and text line detection result

?

# Trend and future

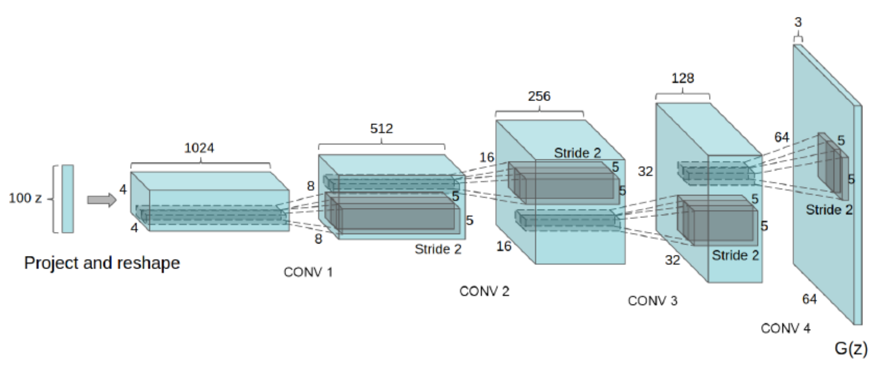
- End to end system
  - Extract all useful features simultaneously if possible (CNN features)
- Multi-task output
  - Human detect text by multiple perspectives
  - If we only do detection task, we can only know a approximate position
  - Once we know more details, more tasks (recognition, segmentation) are used

# Trend and future

- Generative model
  - It is not related much with text detection but also promising
  - Two perspectives for generative model
    - From probabilistic way → $p_{\text{data}}(\mathbf{x})$
    - From generaing way → find x s.t. $p_{\text{data}}(\mathbf{x}) = p$
  - Generative Adversarial Nets

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

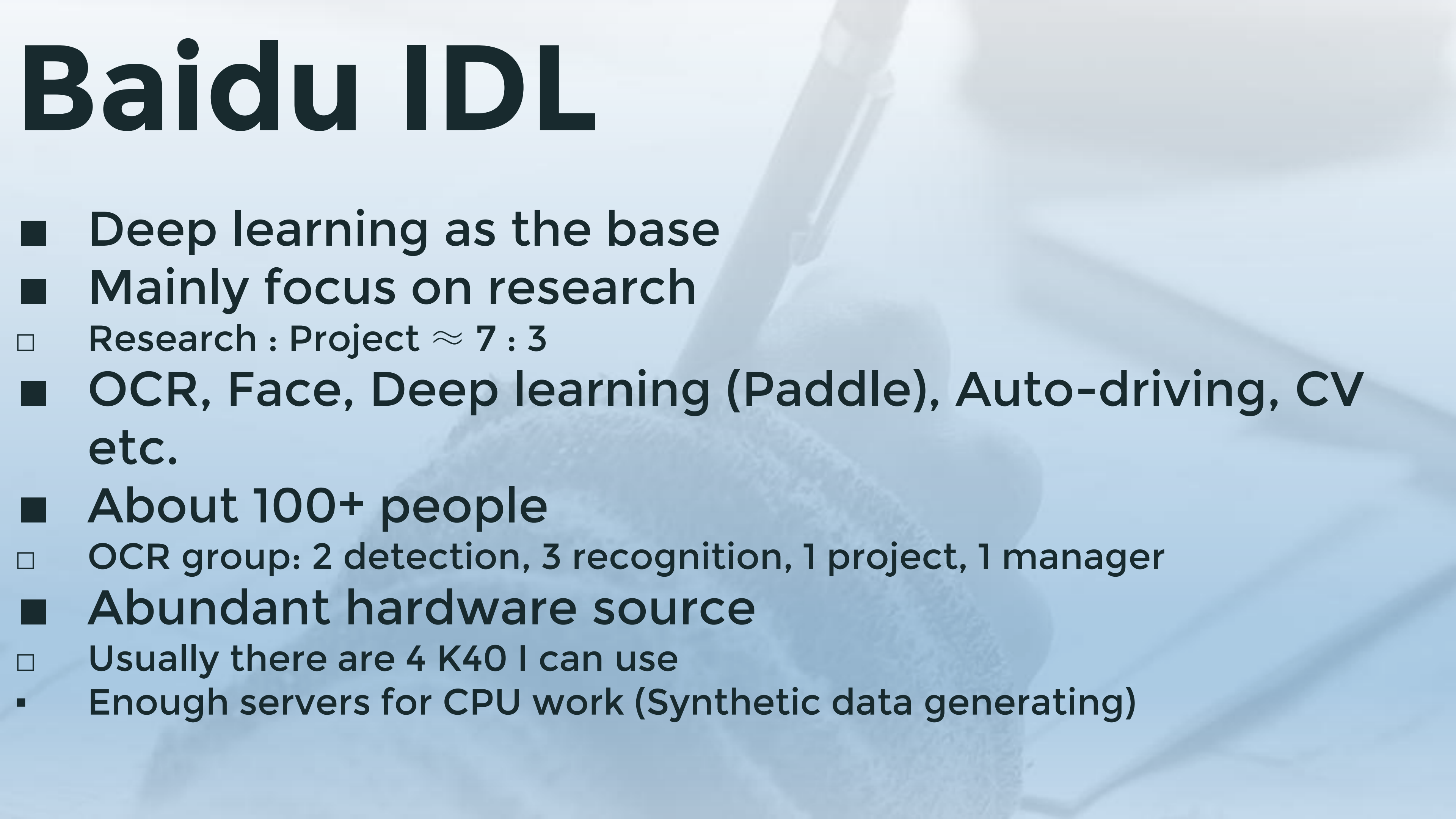
  - Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks



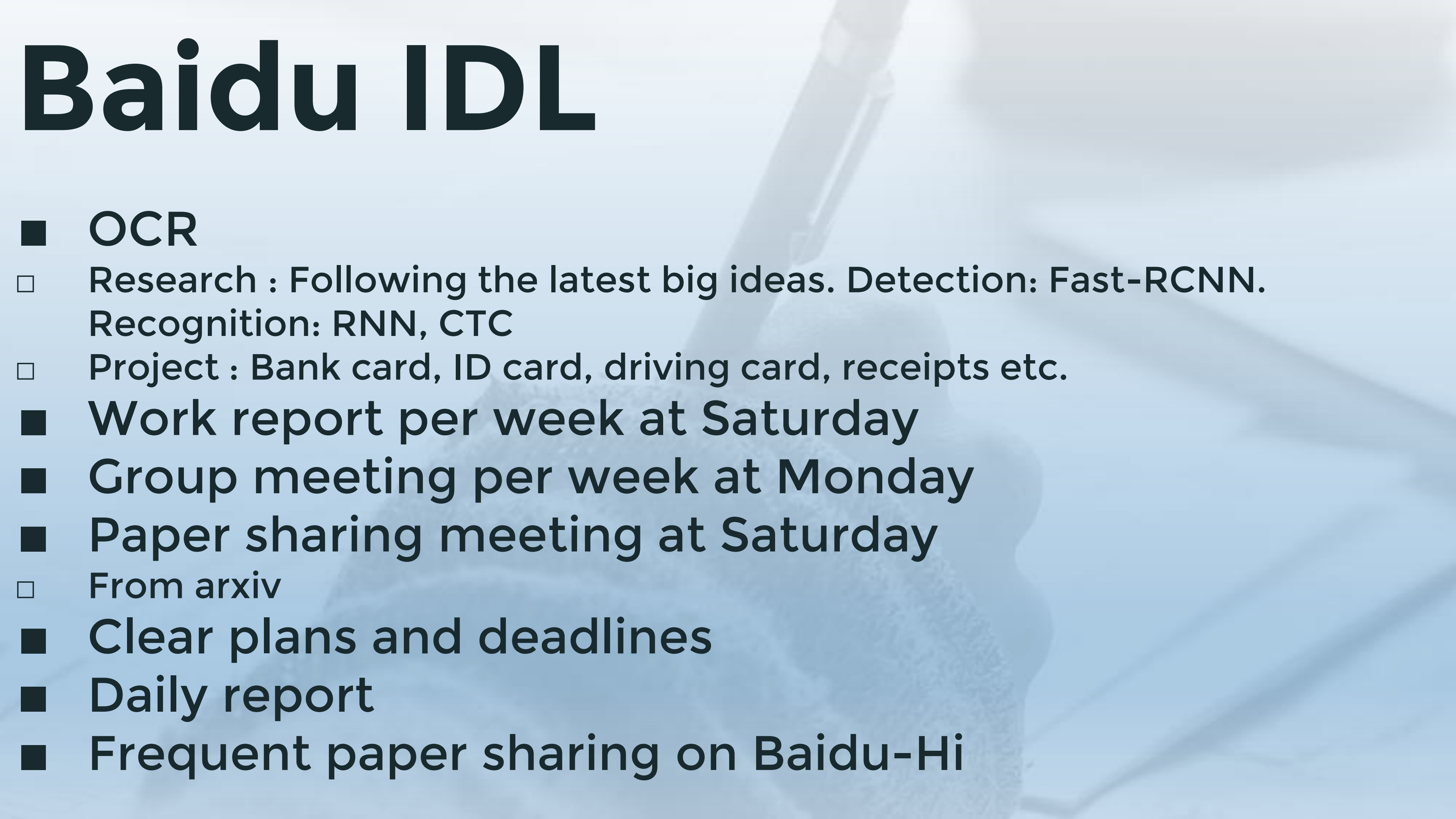
# Trend and future

- Unsupervised and semi-supervised learning
    - How to build a ‘good’ model with less data
    - Supervised data driven is not a merit for deep learning
    - It is not an elegent way using large human-labeling data for training
    - Less training data and synthetic data are recommended

# Baidu IDL



- Deep learning as the base
- Mainly focus on research
  - Research : Project ≈ 7 : 3
- OCR, Face, Deep learning (Paddle), Auto-driving, CV etc.
- About 100+ people
  - OCR group: 2 detection, 3 recognition, 1 project, 1 manager
- Abundant hardware source
  - Usually there are 4 K40 I can use
    - Enough servers for CPU work (Synthetic data generating)



# Baidu IDL

- OCR
  - Research : Following the latest big ideas. Detection: Fast-RCNN. Recognition: RNN, CTC
  - Project : Bank card, ID card, driving card, receipts etc.
- Work report per week at Saturday
- Group meeting per week at Monday
- Paper sharing meeting at Saturday
  - From arxiv
- Clear plans and deadlines
- Daily report
- Frequent paper sharing on Baidu-Hi

# What I have learned

- Technique is only one part
- We are already late, but not too late
- How many papers should I read
- How fast should I realize an idea
- Good research habit
  - Daily plan
  - Weekly plan
  - Deadline
- More communication
  - Know what others are doing
- More idea sharing

# THANKS!

Any questions?