CNN for CV

AI for CV Group
2019

Week 1.
What Is The Course

Compulsory Course:

- CNN for CV

Optional Courses for CV:

- CV in Self-Driving Cars
- Face Recognition
- GAN
- 工程能力提升

Outline:

I. Course Introduction

A. CNN? & CV?
B. Course Schedule

II. Sth Need To Mention

C. Notices Of The Course
D. Coding Preparation
E. Projects

III. Image Introduction

F. Low Level Image Processing (Part I)

I. Course Introduction

I. Course Introduction

A. CNN? & CV?

C: Computer

V: Vision

C: Convolutional

N: Neural

N: Network

I. Course Introduction

A. CNN? & CV?

A1. CNN?

I. Course Introduction

A. CNN? & CV?

A1. CNN?

Others???????

Deep Learning?

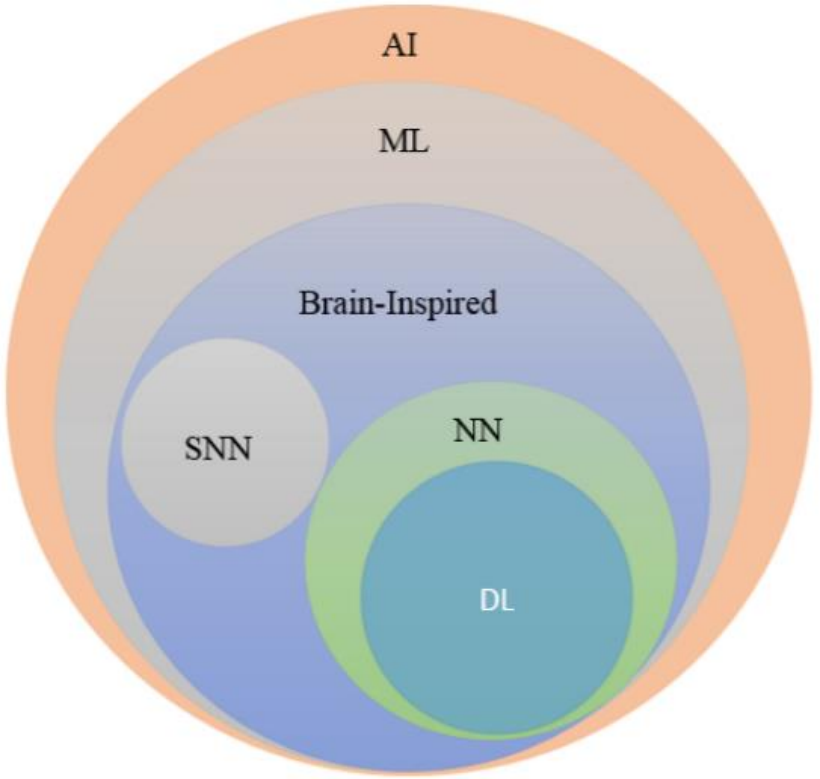
Machine Learning?

Artificial Intelligence?

I. Course Introduction

A. CNN? & CV?

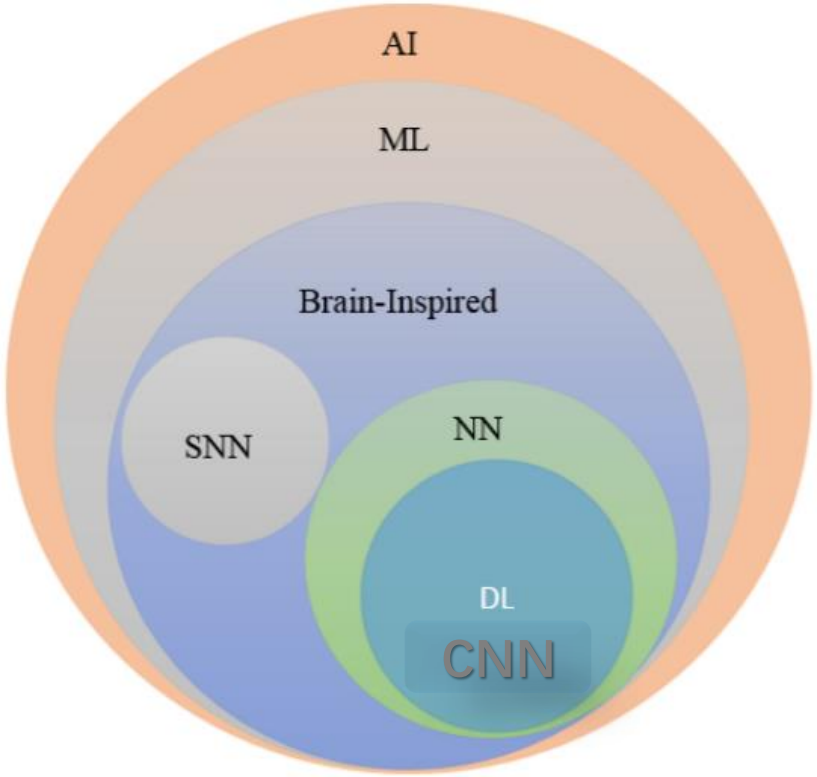
A1. CNN!



I. Course Introduction

A. CNN? & CV?

A1. CNN!

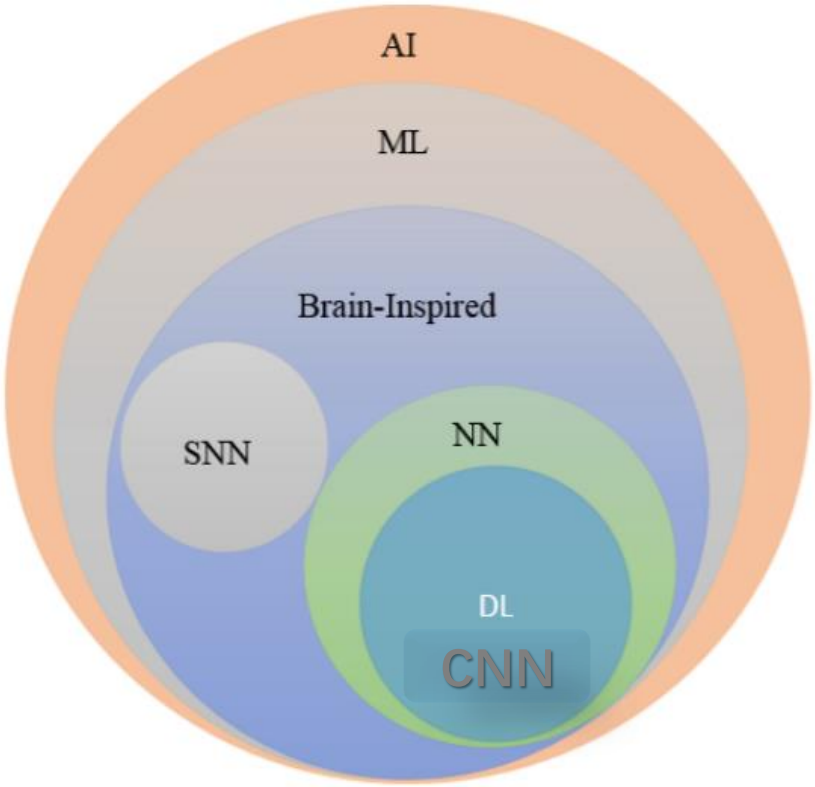


CNN:
A method

I. Course Introduction

A. CNN? & CV?

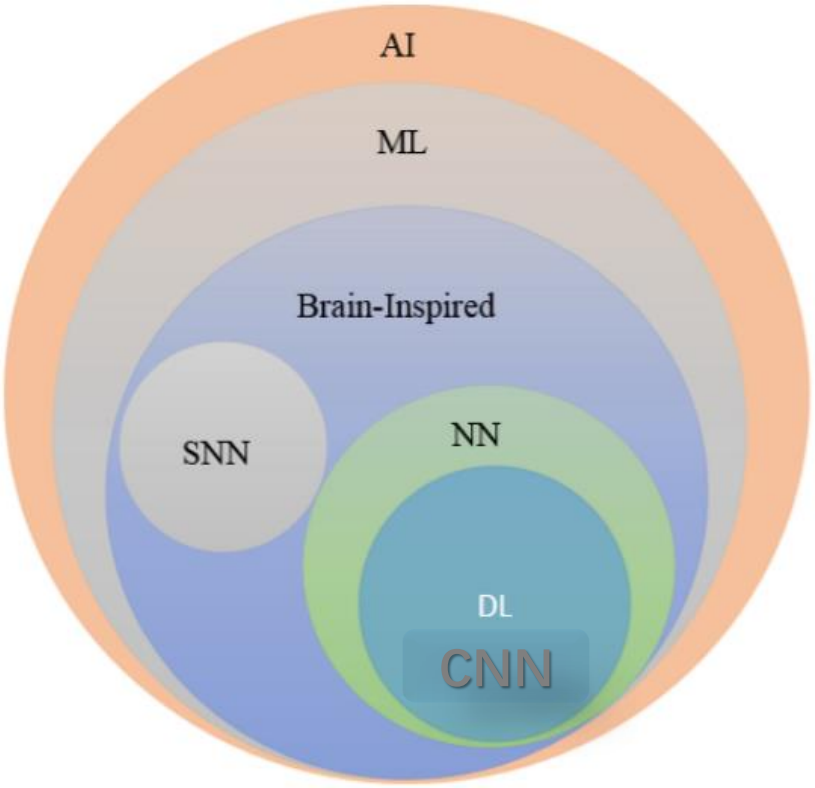
A1. CNN-What's Inside?

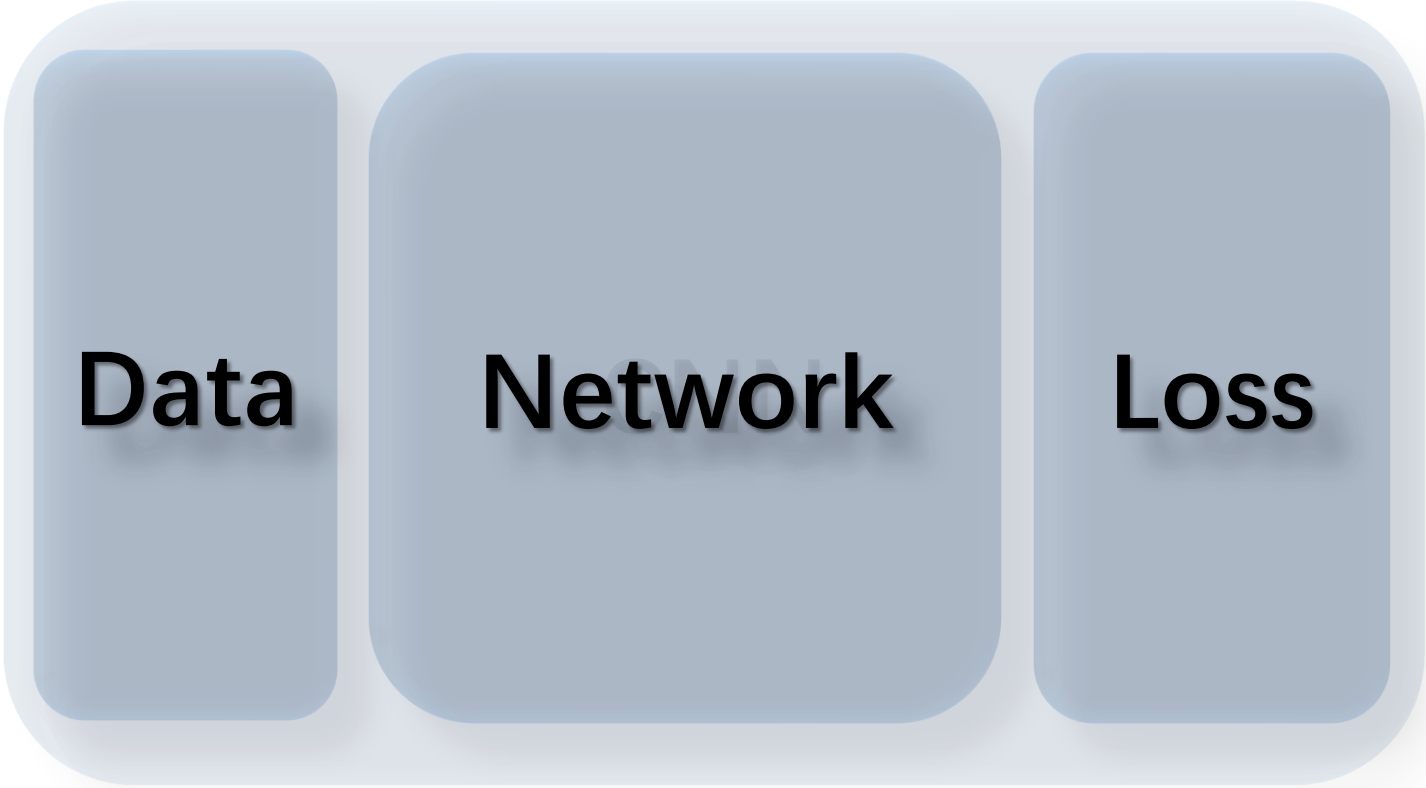


I. Course Introduction

A. CNN? & CV?

A1. CNN-What's Inside?

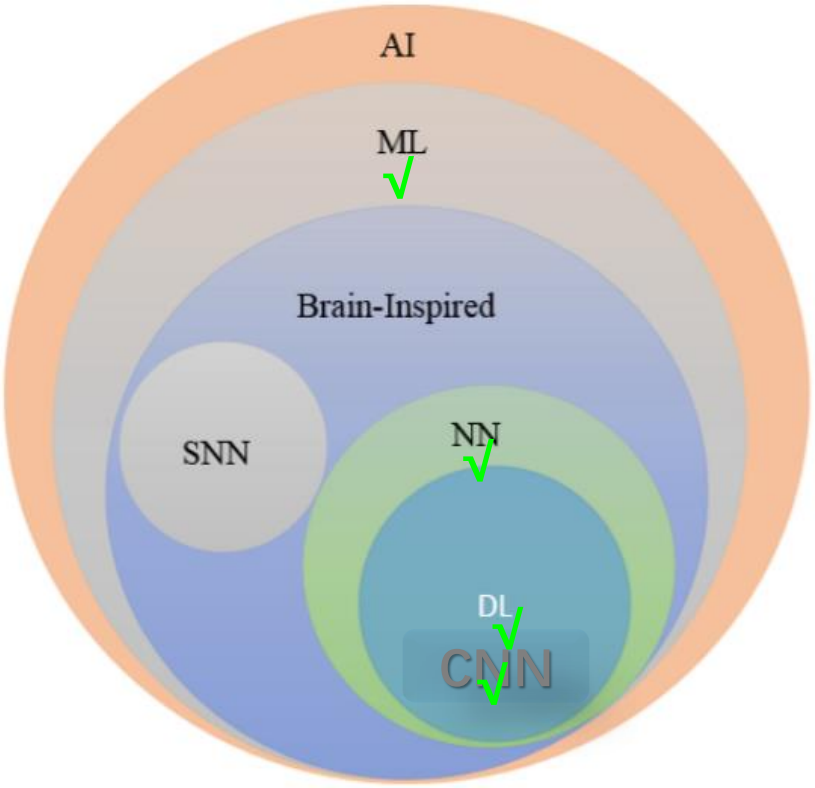




I. Course Introduction

A. CNN? & CV?

A1. CNN-What's Inside?



Data √

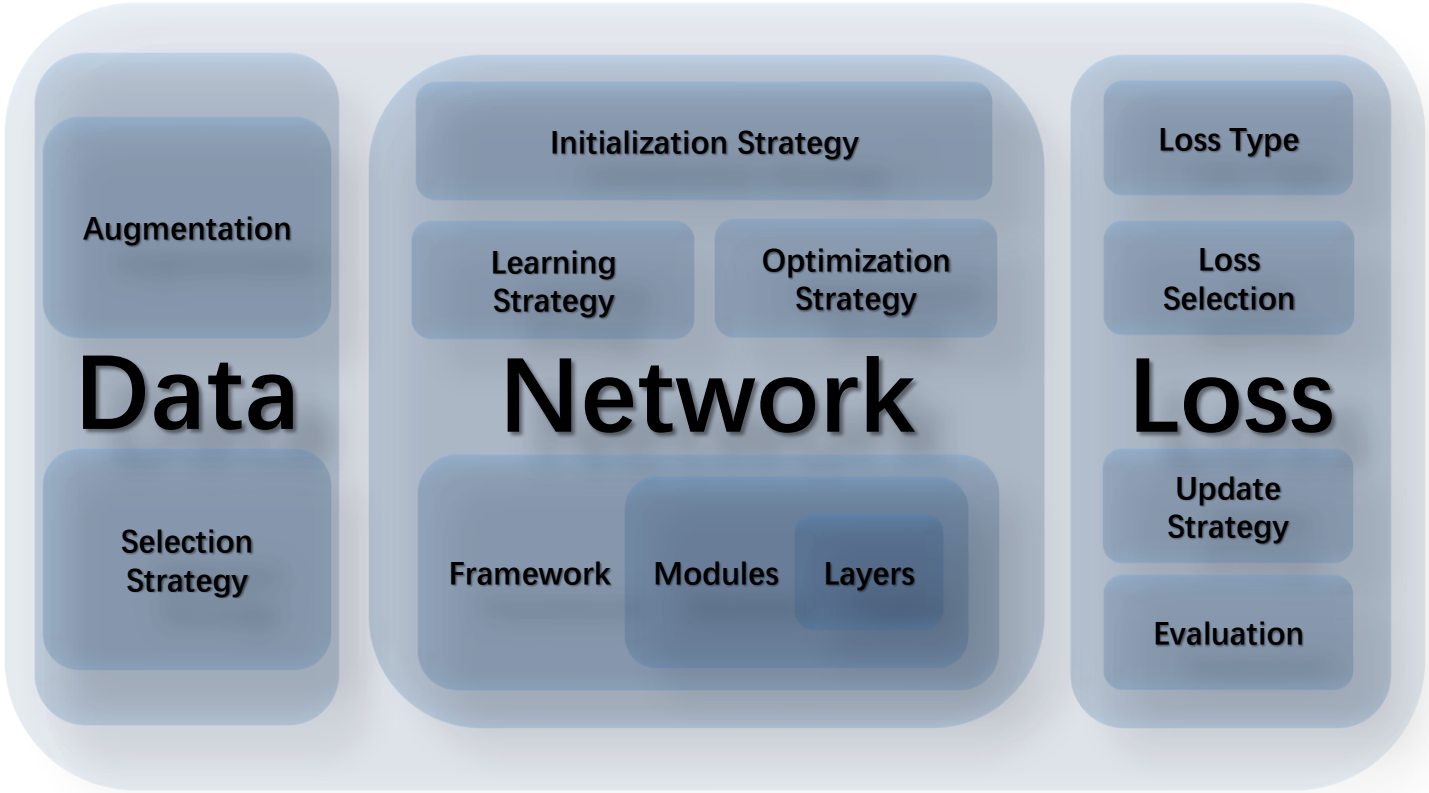
Network √

Loss √

I. Course Introduction

A. CNN? & CV?

A1. CNN-What's Inside?



I. Course Introduction

A. CNN? & CV?

A2. CV?

I. Course Introduction

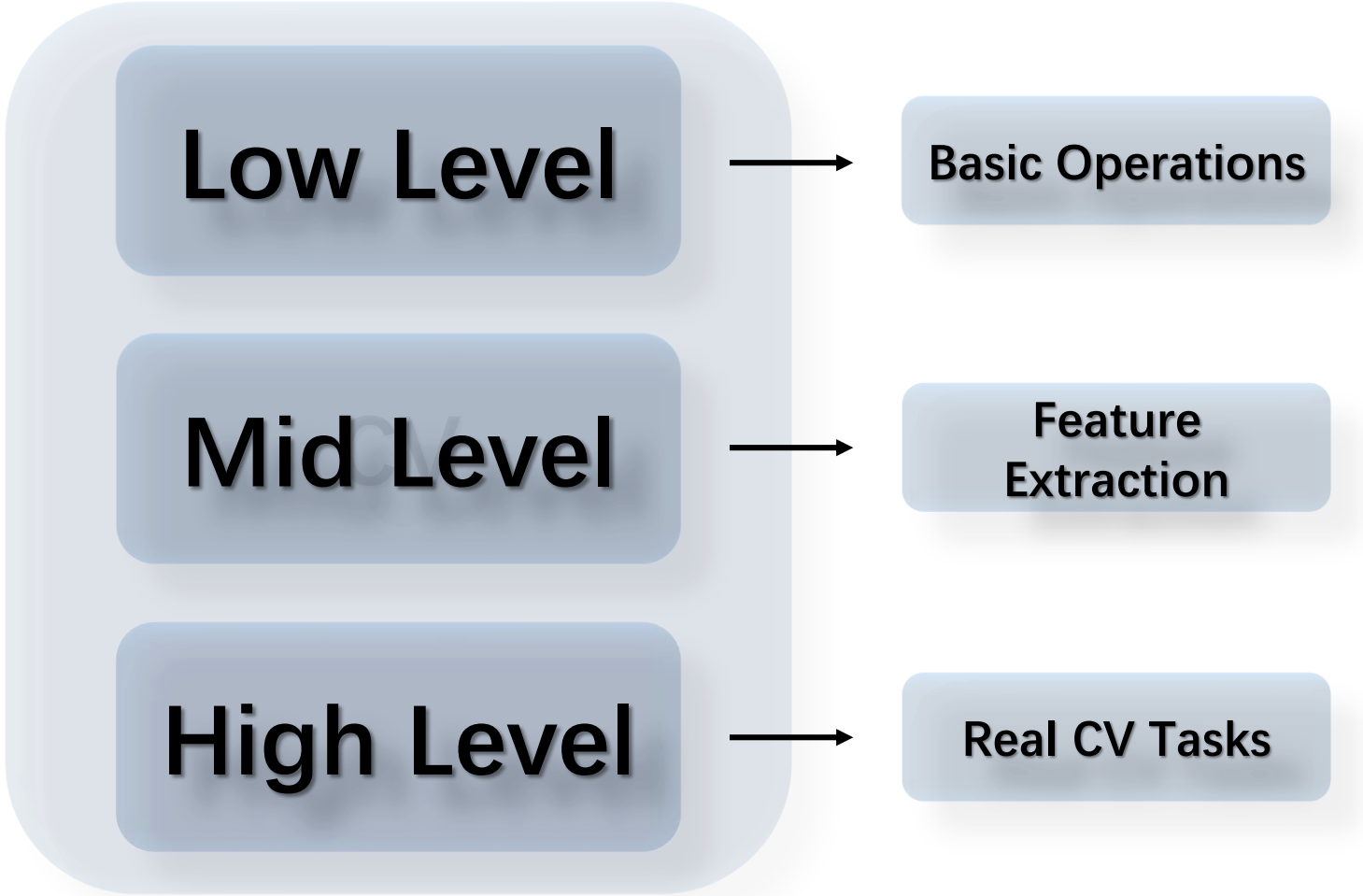
A. CNN? & CV?

A2. CV!

I. Course Introduction

A. CNN? & CV?

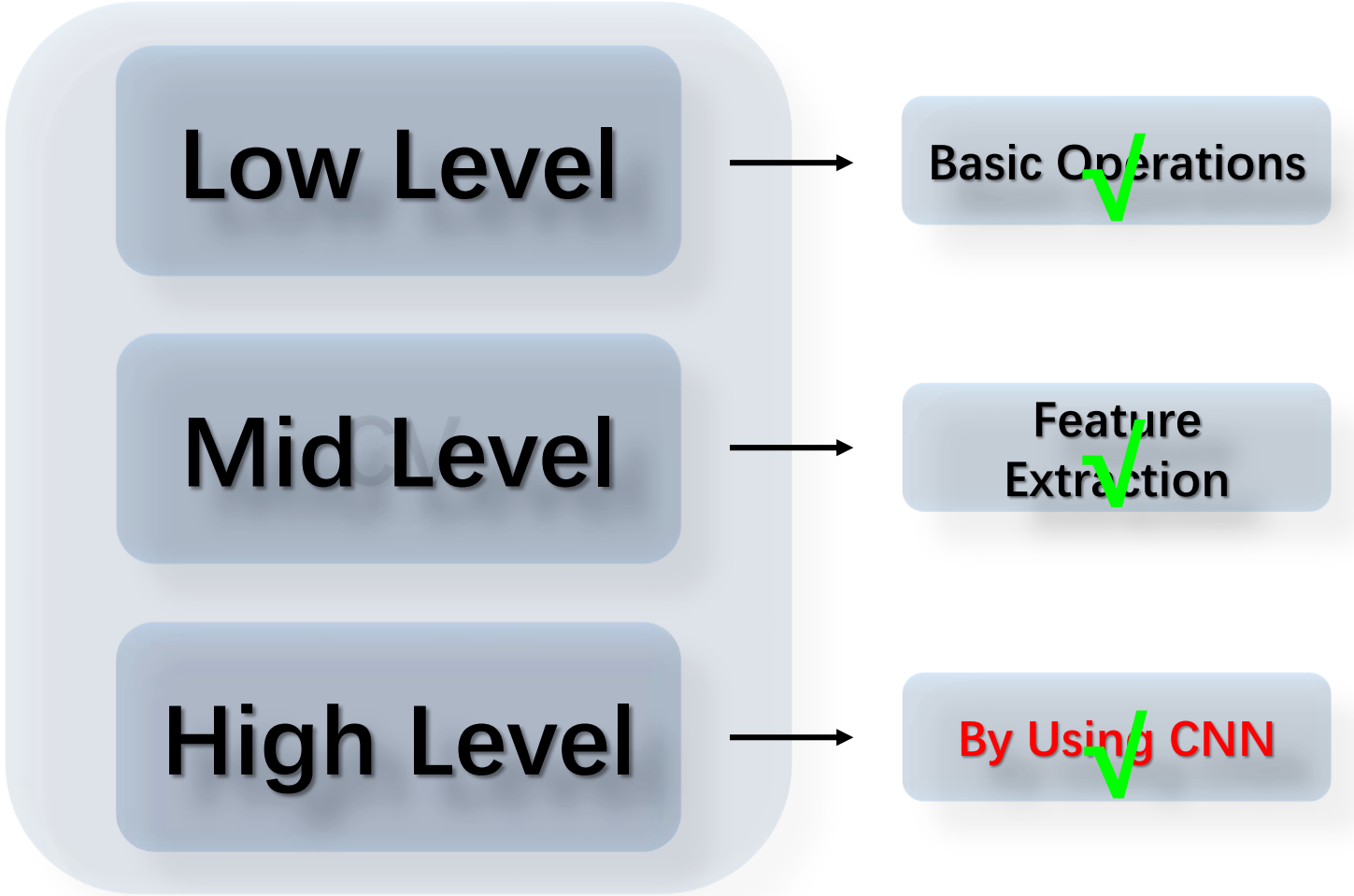
A2. CV!



I. Course Introduction

A. CNN? & CV?

A2. CV!



I. Course Introduction

A. CNN? & CV?

A2. CV!

What Can It Do

I. Course Introduction

A. CNN? & CV?

A2. CV!

I. Course Introduction

A. CNN? & CV?

A2. CV!

I. Course Introduction

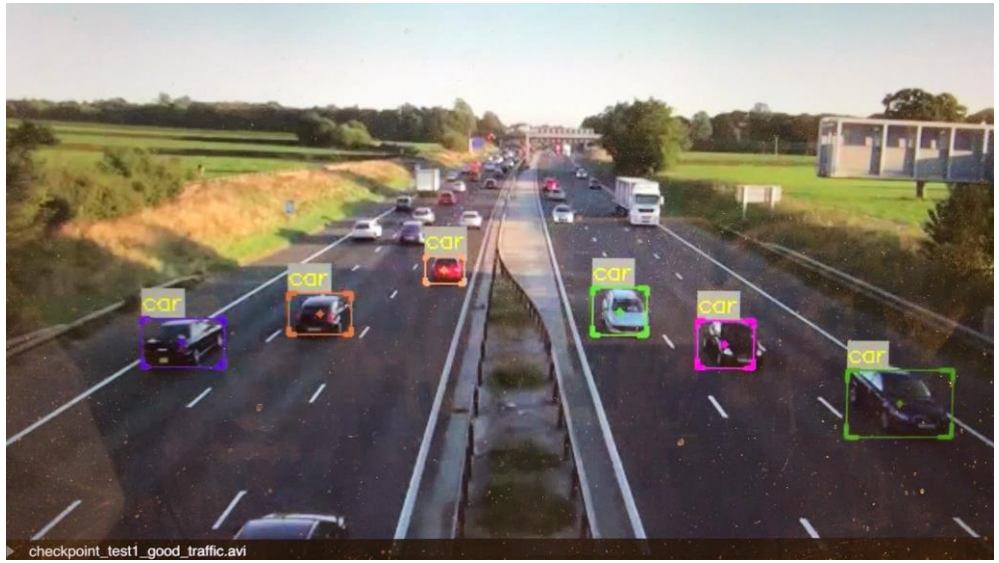
A. CNN? & CV?

A2. CV!

I. Course Introduction

A. CNN? & CV?

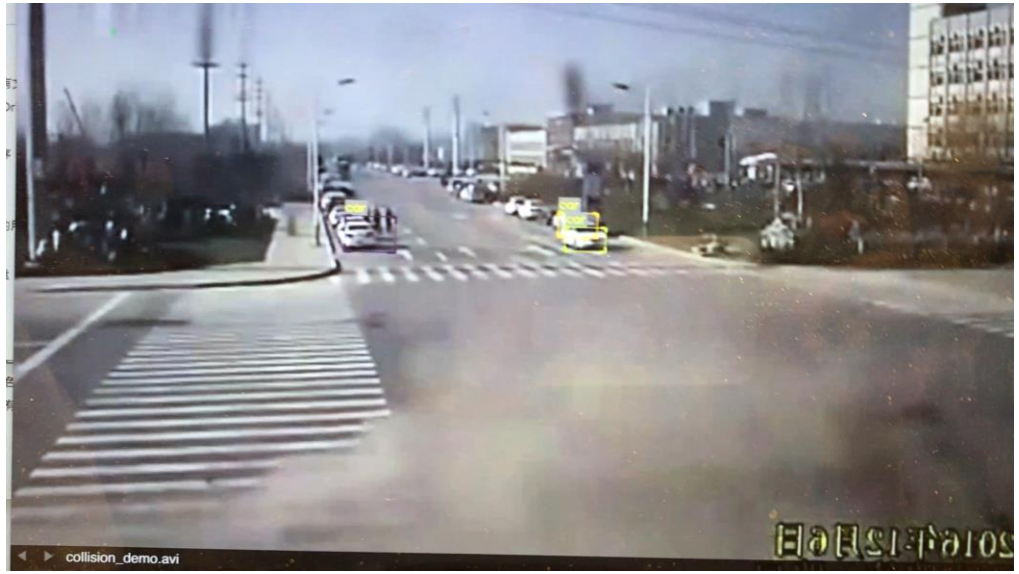
A2. CV!



I. Course Introduction

A. CNN? & CV?

A2. CV!



I. Course Introduction

A. CNN? & CV?

A2. CV!

Fantastic Transformation

Style Transfer

Image Generation

2D To Depth

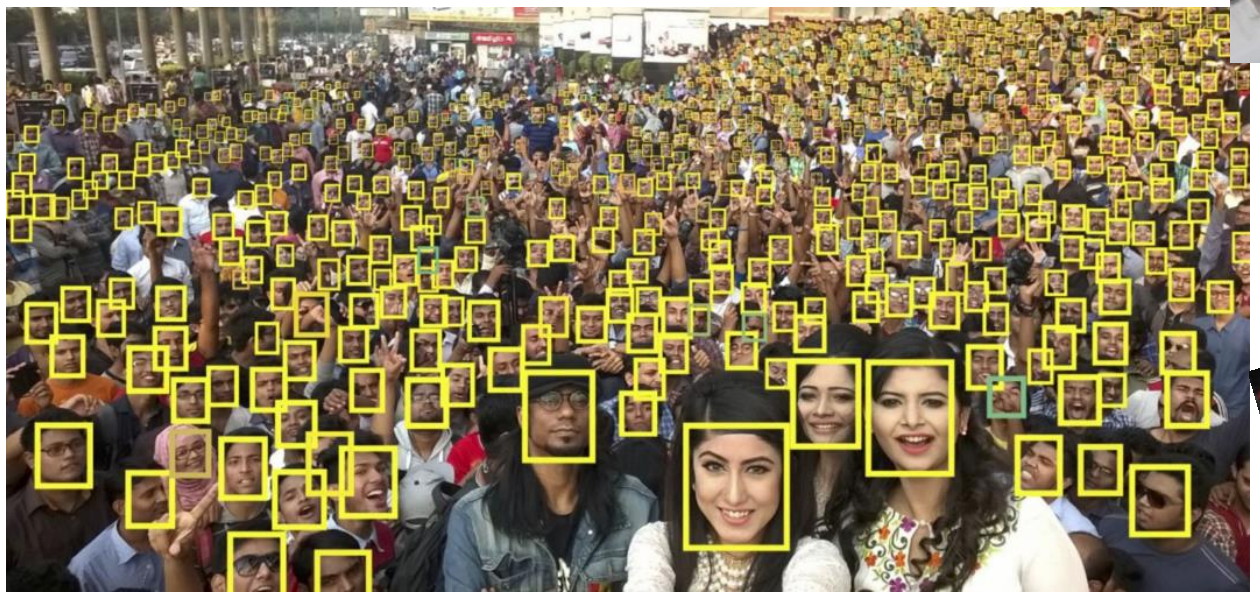
3D objects from 2D

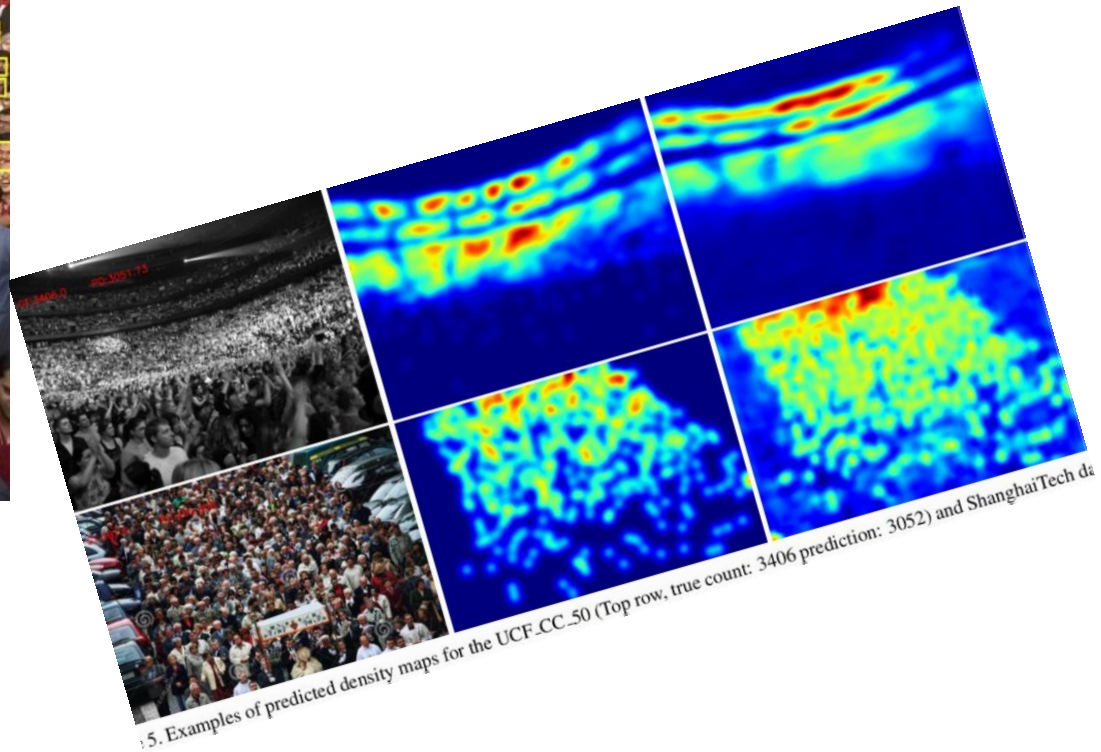
Slam

I. Course Introduction

A. CNN? & CV?

A2. CV!





5. Examples of predicted density maps for the UCF_CC_50 (Top row, true count: 3406 prediction: 3052) and ShanghaiTech da

I. Course Introduction

A. CNN? & CV?

A3. CV-CNN?

I. Course Introduction

A. CNN? & CV?

A3. CV-CNN!

Research:

AutoML
Acceleration
New Applications
New Structure
Refinement
……

Applications:

Image/Video Classification
Object Detection
Keypoint Detection
Recognition
Segmentation
Voxel
Tracking
2D-3D/3D-2D
Image Captioning
Image Transfer
Mixed Inputs
GAN
Slam
……

Engineering:

Modified Models
Light Models
Accelerating Algorithm
Fixed Point
"Applicationized"
Implement Applications
Hardware/Chips
……

I. Course Introduction

A. CNN? & CV?

A3. CV-CNN!

Research:

AutoML
Acceleration
New Applications
New Structure
Refinement
……

Applications:

Image/Video Classification
Object Detection
Keypoint Detection
Recognition
Segmentation
Voxel
Tracking
2D-3D/3D-2D
Image Captioning
Image Transfer
Mixed Inputs
GAN
Slam
……

Engineering:

Modified Models
Light Models
Accelerating Algorithm
Fixed Point
"Applicationized"
Implement Applications
Hardware/Chips
……

I. Course Introduction

B. Course Schedule

CV Fundamental I

CV Fundamental II

ML Fundamental I

ML Fundamental II

ML Fundamental III

CNN Fundamental I

CNN Fundamental II

CNN Fundamental III

Classification

Detection I

Detection II

Image Transfer

Backup

Coding Preparation I

Coding Preparation II

Projects Show

GAN I-Theo.

GAN II-Theo.

GAN III-App.

GAN IV-App.

II. Sth Need To Mention

II. Sth Need To Mention

C. Notices Of Course

Platforms:

Trello, Slack, GitHub

Languages:

Python, C++

Frameworks:

Caffe, Caffe2, Tensorflow, CNTK, Paddlepaddle, MXNet, PyTorch

slack

GitHub

II. Sth Need To Mention

C. Notices Of Course

Platforms:

Trello, Slack, GitHub

Languages:

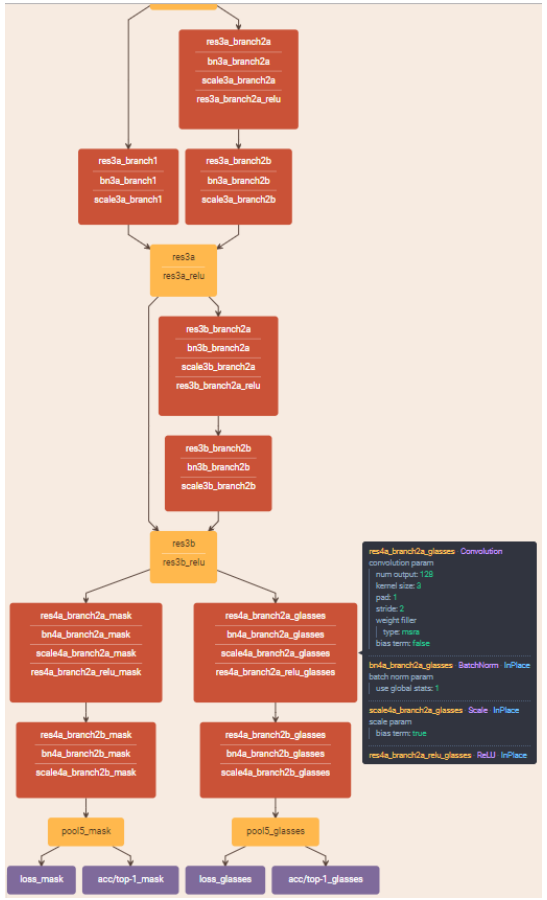
Python, C++

Frameworks:

Caffe, Caffe2, Tensorflow, CNTK, Paddlepaddle, MXNet, **PyTorch**

II. Sth Need To Mention

C. Notices Of Course



Caffe ⟶ Caffe2
(UC Berkeley) (Facebook)

PaddlePaddle
(Baidu)

Torch ⟶ PyTorch
(NYU/Facebook) (Facebook)

MXNet
(Amazon)

Theano ⟶ TensorFlow
(U Montreal) (Google)

CNTK
(MS)

II. Sth Need To Mention

D. Coding Preparation

- **Coding Test**
- **Extremely Critical**
- **200 / 300+**
- **40min, 4 / 5, medium**
- **Strategy / Material**
- **Lintcode / Leetcode / 九章算法**

II. Sth Need To Mention

D. Coding Preparation

计算机视觉算法
岗位职责：
1. 负责以机器/深度学习为基础的高级驾驶辅助系统物体检测算法，包括模型训练和优化。
2. 负责计算机视觉、深度学习相关方向的技术难点攻关与前瞻研究，包括人脸检测/识别、车辆检测/追踪等。
任职要求：
1. 熟悉机器学习（Boosting等）和深度学习（CNN等），有caffe，tensorflow，mxnet等至少一个深度学习框架使用经验。
2. 熟悉并实践过物体检测算法，如SSD、Fast-RCNN等，熟练掌握训练、评测、调参等过程。
3. 优秀的Python/C/C++编程能力，良好的代码风格。熟练Linux操作，掌握Git开发流程，了解Cmake、GCC等。

算法工程师（上海/北京）
工作内容
1.智能算法研究，快速实现前沿工作，创新理论框架
2.根据客户场景，完成算法设计，实现和调优，推进算法落地
3.负责计算机视觉、自然语言理解、深度学习算法等领域的研发工作，保持算法在工业界和学术界的领先
4.追踪技术前沿，判断最新技术对产品的影响
职位要求
1.熟悉Linux操作系统，熟悉C++及Python/Bash开发，本科及本科学历以上
2.熟悉CNN等深度学习技术，有caffe/tensorflow/pytorch使用经验更佳
3.对机器学习、计算机视觉、自然语言理解等领域有强烈好奇心
4.有优秀的分析问题和解决问题的能力，乐于挑战尖端研究课题
5.良好的沟通能力和团队合作精神。

语义Slam算法
职位描述：
1. 开发基于语义感知的SLAM系统；
2. 开发交通目标的检测&分割系统，包括车道线、路沿、标识牌等；
3. 多批次数据的SLAM联合优化。
职位要求：
1. 具有扎实的数学基础，精通多视几何、非线性优化等；
2. 具有SFM/SLAM等算法开发和系统实现经验，具有视觉目标检测场景分割经验；
3. 熟悉OpenCV&Ceres&g2o及TF/MxNet/Caffe等库；
4. 扎实的C/C++/python等语言知识及熟练使用常用的数据结构与算法。
优先项：
1. 有相应论文或可展示作品的优先；

计算机视觉算法
岗位职责：
1．负责以机器/深度学习为基础的高级驾驶辅助系统物体检测算法，包括模型训练和优化。
2．负责计算机视觉、深度学习相关方向的技术难点攻关与前瞻研究，包括人脸检测/识别、车辆检测/追踪等。
任职要求：
1．熟悉机器学习（Boosting等）和深度学习（CNN等），有caffe，tensorflow，mxnet等至少一个深度学习框架使用经验。
2．熟悉并实践过物体检测算法，如SSD、Fast-RCNN等，熟练掌握训练、评测、调参等过程。
3．优秀的Python/C/C++编程能力，良好的代码风格。熟练Linux操作，掌握Git开发流程，了解Cmake、GCC等。

语义Slam算法
职位描述：
1．开发基于语义感知的SLAM系统；
2．开发交通目标的检测&分割系统，包括车道线、路沿、标识牌等；
3．多批次数据的SLAM联合优化。
职位要求：
1．具有扎实的数学基础，精通多视几何、非线性优化等；
2．具有SFM/SLAM等算法开发和系统实现经验，具有视觉目标检测场景分割经验；
3．熟悉OpenCV&Ceres&g2o及TF/MxNet/Caffe等库；
4．扎实的C/C++/python等语言知识及熟练使用常用的数据结构与算法。
优先项：
1．有相应论文或可展示作品的优先；

II. Sth Need To Mention

D. Coding Preparation

计算机视觉算法工程师(智能物联网)
工作职责:
1、负责计算机视觉、深度学习相关的技术系统与产品的研发工作;
2、负责计算机视觉、深度学习相关方向的技术难点攻关与前瞻研究;
3、负责算法计算性能优化，并推动其上线应用。
任职资格:
1、创造性思维，富有想象力，有推进人工智能的理想和使命感;
2、在深度学习、统计机器学习、计算机视觉、最优化方法等方面有较深入的研究;
3、熟悉物体（人体、人脸、通用目标）检测、跟踪与识别的基本算法;
4、较强的逻辑思维能力以及算法实现能力;
5、具有良好的沟通能力和团队合作精神。

计算机视觉/深度学习算法研究员
岗位职责:
1、负责计算机视觉和深度学习算法的研发，研发下属课题中的一项或多项，包括不限于:物体检测、跟踪、识别、分类、语义分割、深度估计、图像处理、视频结构化分析、强化学习相关算法等。
2、推动研发的算法在实际场景应用领域的性能优化和落地。
要求:
1、在计算机视觉、机器学习和深度学习方面有深入研究。
2、算法基础扎实，有较强的算法实现能力，熟悉掌握C/C++编程，熟悉Shell/Python/Matlab编程。
3、有较强的研究能力者优先，比如在国际顶尖会议或期刊（包括不限于CVPR、ICCV、ECCV、NIPS等）上发表过论文。
4、有很强的代码能力者优先，比如获得过ACM或商业代码竞赛荣誉，或代码开源在GitHub上有较大影响者优先。
5、有很强的比赛经验者或一些重要数据集的排行榜上靠前的优先，比如ActivityNet竞赛、ImageNet，FDDB等。

II. Sth Need To Mention

D. Coding Preparation

试用期期间工资为 转正工资的 100% ，试用期结束之后税前 肆万元 人民币每月

II. Sth Need To Mention

E. Projects

II. Sth Need To Mention

E. Projects

THE MOST INPORTANT THING

understand what you want

II. Sth Need To Mention

E. Projects

- **Do Sth New:**
  a. Sth really new
  b. Less people do
  c. Structure / Data / Training
  d. App

II. Sth Need To Mention

E. Projects

- **Do Sth New:**
  a. Sth really new
  b. Less people do
  c. Structure / Data / Training
  d. App

- **Self Driven**
  a. Group
  b. Presentation: Slides & Talk

II. Sth Need To Mention

E. Projects

- **Advices:**
  3 projects: 1 + 1 + 1
  1: classical CV (3 weeks)
  1: simple CNN (1-2 weeks) – like multi-classification
  1: real task (rest time)

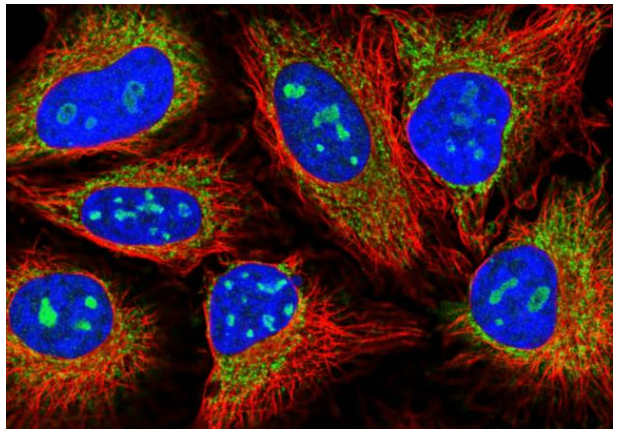
II. Sth Need To Mention

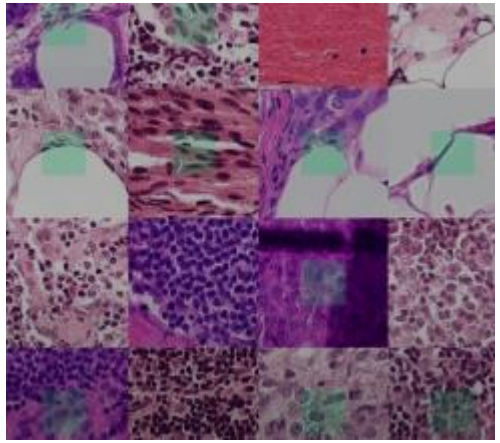
E. Projects

- **Tiny Projects:**

Kaggle:

- **Cancer Detection:** link1
- **Cactus Identification:** link1
- **Protein Atlas Image Classification:** link1

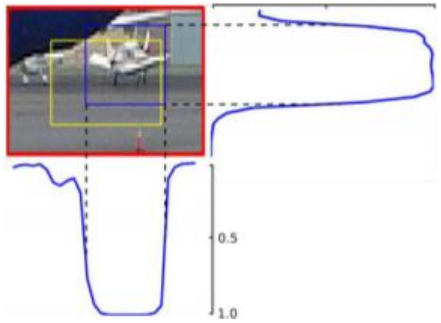


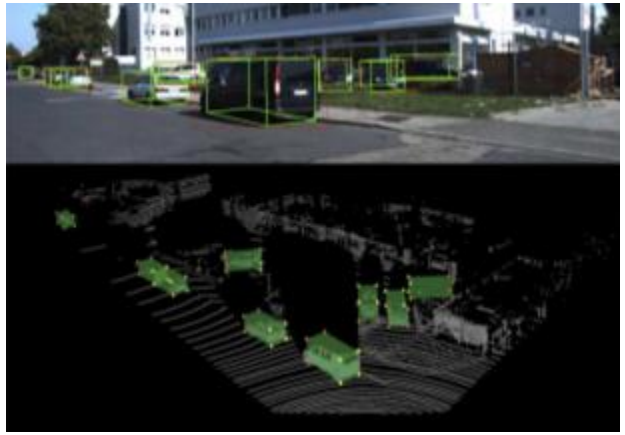


II. Sth Need To Mention

E. Projects

- **Projects:**
  detection:
  - **Detection with rotated bbx: link1, link2**
    (More Accurate: Orientation)
  - **LocNet: link1, link2**
    (More Accurate: BBX)
  - **PointFusion: link1** **3D BBX: link1**
    (More Accurate: 2D-3D)

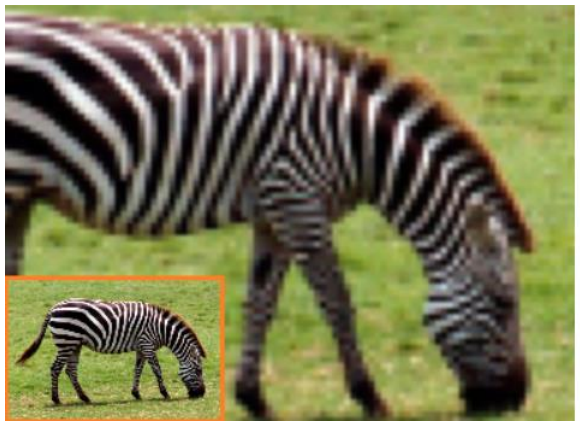




II. Sth Need To Mention

E. Projects

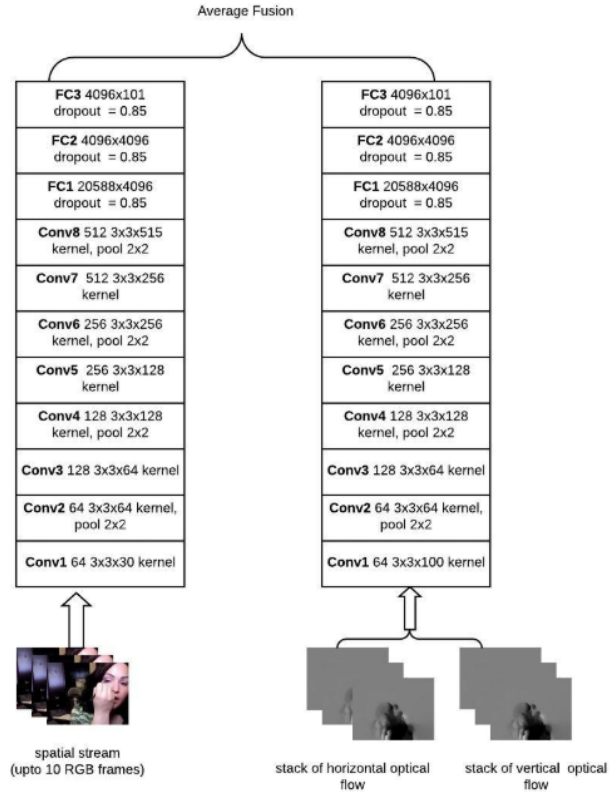
- **Projects: Image Transferring:**
  - **Reflection Removal: link1**
  - **Super Resolution: link1, link2**
  - **Face Frontalization: link1**
  - **Add / Remove Sth: link1**

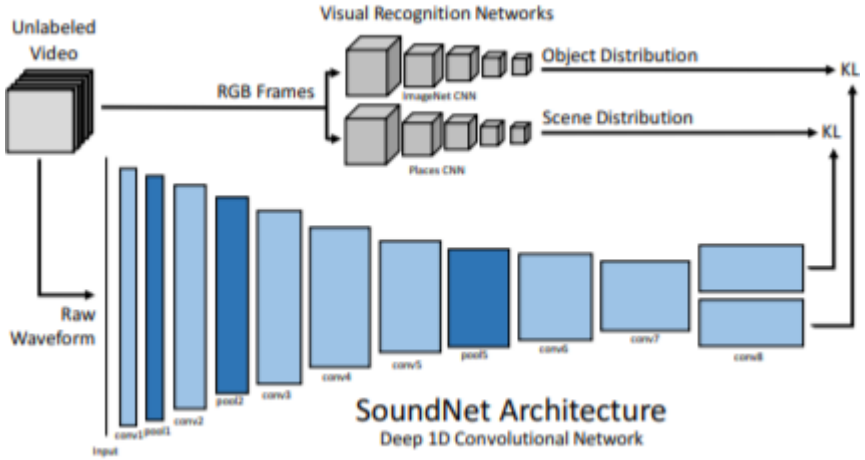


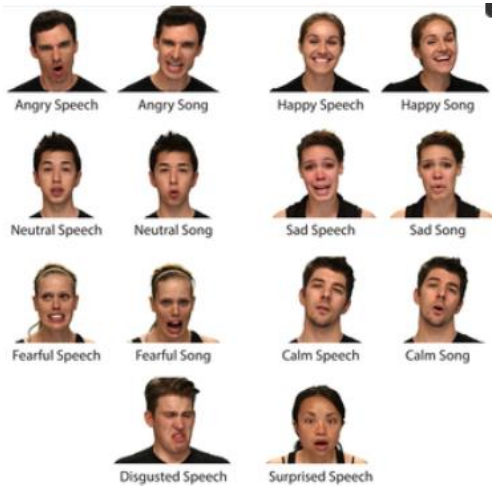
II. Sth Need To Mention

E. Projects

- **Projects: Multi-Inputs/outputs:**
  - **Video->Sound: link1, link2**
  - **Video + Sound -> Classification: link1**
  - **2-Stream Videos-> Classification: link1**







II. Sth Need To Mention

E. Projects

- **Projects:**
  Others:

Image/Video Classification
Object Detection
Keypoint Detection
Face Detection/Recognition
Segmentation
Voxel
Tracking
2D-3D/3D-2D
Image Captioning
Image Transfer
Mixed Inputs
GAN
Slam
……

II. Sth Need To Mention

Summary

1. **Find your field AS SOON AS POSSIBLE**
2. **Be FRIENDLY with recruiters**
3. **REALLY CARE about job description and FORGET the contents**
4. **CODE MORE, THINK MORE, READ MORE**
5. **CVPR, ECCV, ICCV, NIPS / PAMI / SIGGRAPH / arXiv / Kaggle**

III. Image Introduction

III. Image Introduction

F. Low Level Image Processing

Basic Operations I:

- Read in (Gray or Color) / Show out
- Print out an image
- Image data type/shape/crop
- Channels / Spaces
- Change color (Assignment)
- Gamma Correction
- Histogram (Thinking: adaptive & align)

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

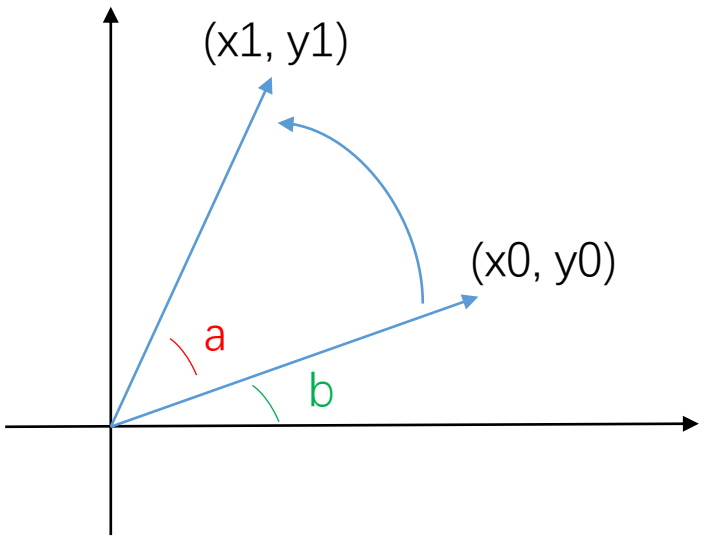
- Similarity Transform
- Affine Transform
- Perspective Transform

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Rotation



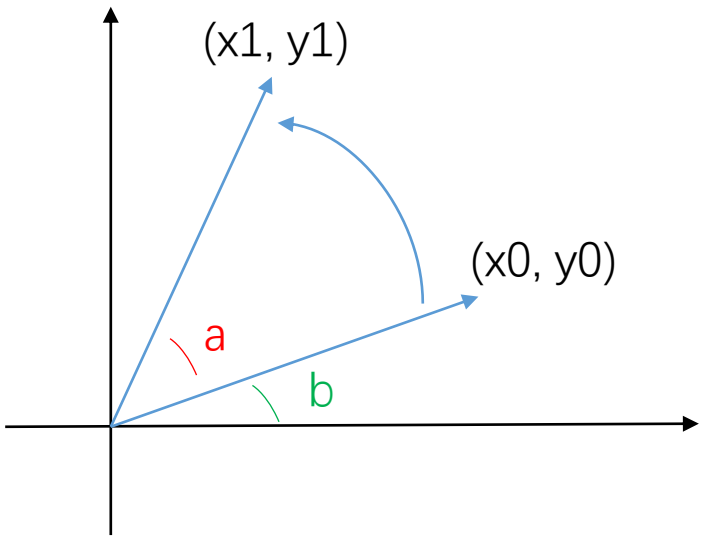
$$\overrightarrow{x_1} = \boldsymbol{R}\overrightarrow{x_0} = \begin{bmatrix} \cos a & -\sin a \\ \sin a & \cos a \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Rotation+Translation



$$\overrightarrow{x_1} = \boldsymbol{R}\overrightarrow{x_0} = \begin{bmatrix} \cos a & -\sin a \\ \sin a & \cos a \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

$$\overrightarrow{x_1} = [\boldsymbol{R}|\boldsymbol{t}]\overrightarrow{x_0} = \begin{bmatrix} \cos a & -\sin a & t_x \\ \sin a & \cos a & t_y \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

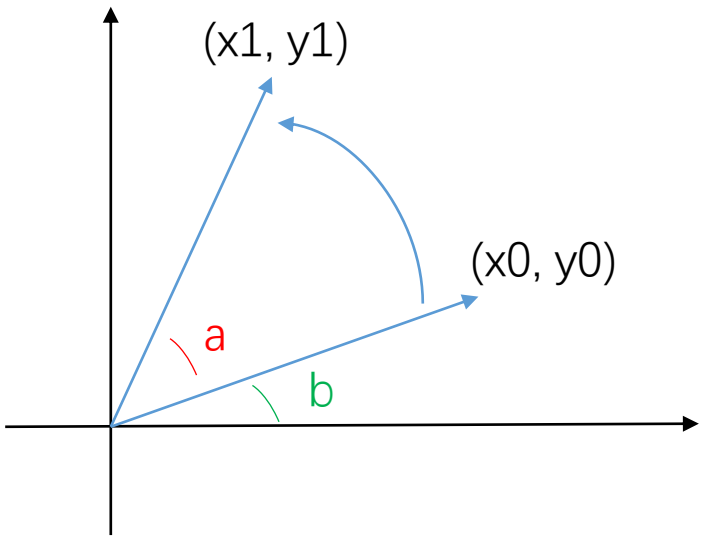
Homogeneous Coordinate (齐次坐标)

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Rotation+Translation+Scale=Similarity Transform



$$\overrightarrow{x_1} = \boldsymbol{R}\overrightarrow{x_0} = \begin{bmatrix} \cos a & -\sin a \\ \sin a & \cos a \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

$$\overrightarrow{x_1} = [\boldsymbol{R}|\boldsymbol{t}]\overrightarrow{x_0} = \begin{bmatrix} \cos a & -\sin a & t_x \\ \sin a & \cos a & t_y \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

Homogeneous Coordinate (齐次坐标)

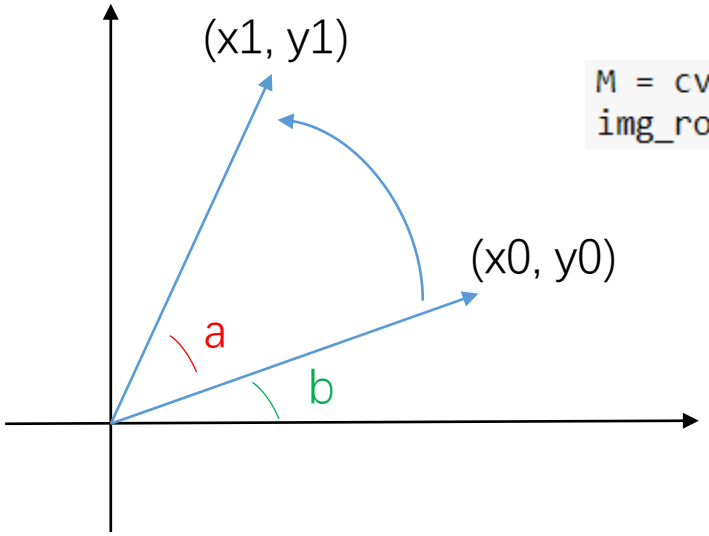
$$\overrightarrow{x_1} = \mathrm{s}[\boldsymbol{R}|\boldsymbol{t}]\overrightarrow{x_0} = \mathrm{s} \begin{bmatrix} \cos a & -\sin a & t_x \\ \sin a & \cos a & t_y \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Similarity Transform



New Centers Angle Scale

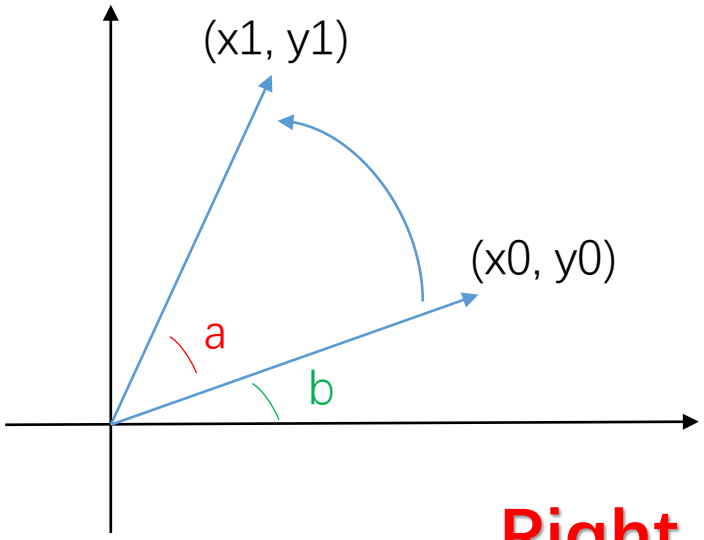
```
M = cv2.getRotationMatrix2D((img.shape[1] / 2, img.shape[0] / 2), 30, 0.5)
img_rotate = cv2.warpAffine(img, M, (img.shape[1], img.shape[0]))
```

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Similarity Transform



1. **Regard (0,0) as base**
2. **Scale**
3. **Rotate around (0,0)**
4. **Translate to the new center**

Right Angle Is STILL Right Angle

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Similarity Transform

Right Angle Is STILL Right Angle

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- **Affine (仿射) Transform**

$$\overrightarrow{x_1} = \boldsymbol{A}\overrightarrow{x_0} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Affine (仿射) Transform

$$\overrightarrow{x_1} = \boldsymbol{A}\overrightarrow{x_0} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

Parallel Lines Are STILL Parallel Lines

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- **Affine (仿射) Transform**

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Affine (仿射) Transform

3 Src Points + 3 Dst Points

```
pts1 = np.float32([[0, 0], [cols - 1, 0], [0, rows - 1]])
pts2 = np.float32([[cols * 0.2, rows * 0.1], [cols * 0.9, rows * 0.2], [cols * 0.1, rows * 0.9]])

M = cv2.getAffineTransform(pts1, pts2)
dst = cv2.warpAffine(img, M, (cols, rows))
```

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Affine (仿射) Transform

3 Non-Collinear Point Pairs determines 1 Affine Transform

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Affine (仿射) Transform

3 Non-Collinear Point Pairs determines 1 Affine Transform

WHY ??

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Affine (仿射) Transform

3 Non-Collinear Point Pairs determines 1 Affine Transform

$$\overrightarrow{x_1} = \boldsymbol{A}\overrightarrow{x_0} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

6 Parameters !

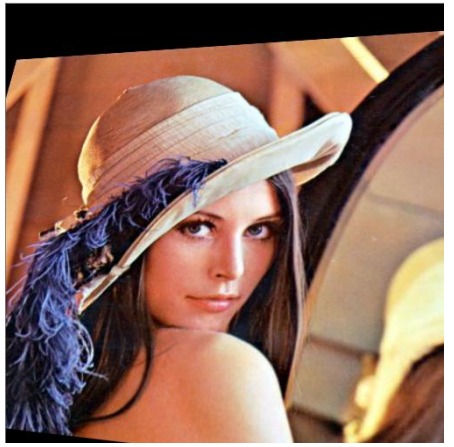
(6 Degree of Freedom)

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Perspective (投影) Transform



III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Perspective (投影) Transform

$$\overrightarrow{\boldsymbol{I_1}} = \boldsymbol{P}\overrightarrow{\boldsymbol{I_0}} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ w_0 \end{bmatrix}$$

Lines Are STILL Lines

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Perspective (投影) Transform

$$\overrightarrow{\boldsymbol{I_1}} = \boldsymbol{P}\overrightarrow{\boldsymbol{I_0}} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ w_0 \end{bmatrix}$$

How many points do we need to solve it?

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Perspective (投影) Transform

$$\overrightarrow{\boldsymbol{I}_1} = \boldsymbol{P}\overrightarrow{\boldsymbol{I}_0} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ w_0 \end{bmatrix}$$

4 !

8 Degree of Freedom !

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Perspective (投影) Transform

$$\overrightarrow{I_1} = P\overrightarrow{I_0} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ w_0 \end{bmatrix}$$

4 !

$||P||^2$ = 1, when calculation

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Perspective (投影) Transform

```
pts1 = np.float32([[x1, y1], [x2, y2], [x3, y3], [x4, y4]])
pts2 = np.float32([[dx1, dy1], [dx2, dy2], [dx3, dy3], [dx4, dy4]])
M_warp = cv2.getPerspectiveTransform(pts1, pts2)
img_warp = cv2.warpPerspective(img, M_warp, (width, height))
```

III. Image Introduction

F. Low Level Image Processing

Basic Operations II:

- Perspective (投影) Transform

Perspective Transform (Particularly for 4 points)

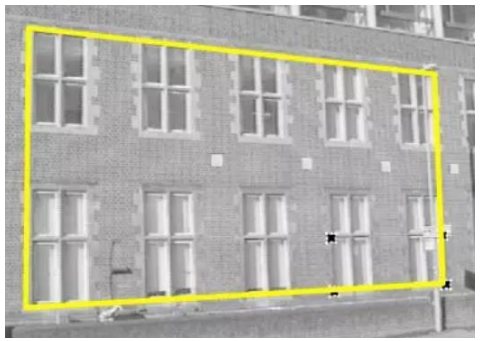
(>= 4 points: Homography 单应性)

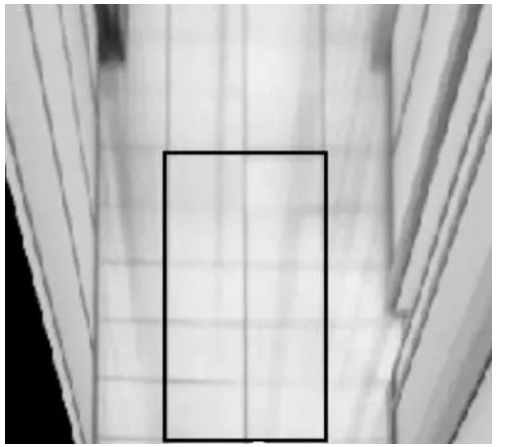
III. Image Introduction

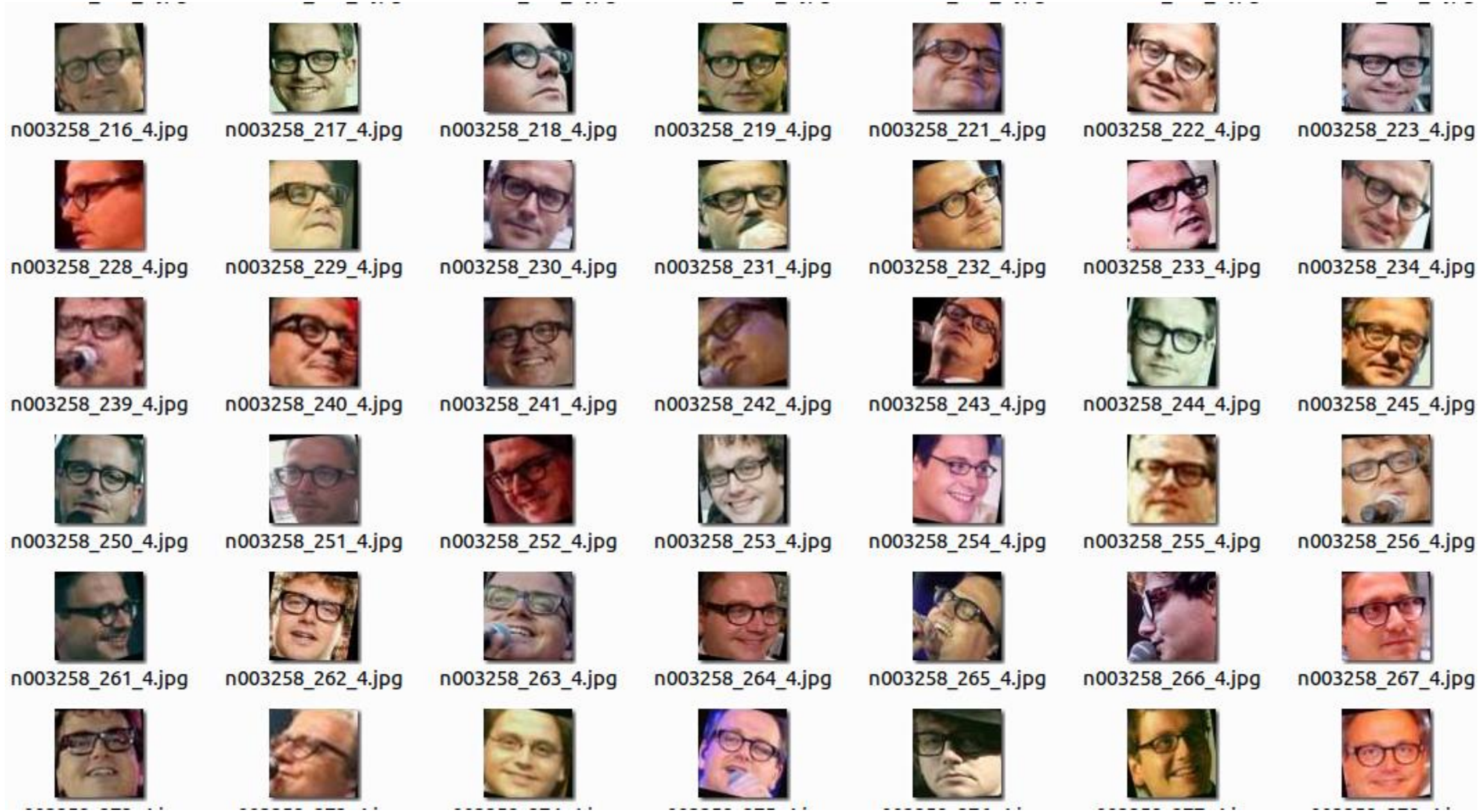
F. Low Level Image Processing

Basic Operations II:

- Perspective (投影) Transform






n003258_216_4.jpg
n003258_217_4.jpg
n003258_218_4.jpg
n003258_219_4.jpg
n003258_221_4.jpg
n003258_222_4.jpg
n003258_223_4.jpg
n003258_228_4.jpg
n003258_229_4.jpg
n003258_230_4.jpg
n003258_231_4.jpg
n003258_232_4.jpg
n003258_233_4.jpg
n003258_234_4.jpg
n003258_239_4.jpg
n003258_240_4.jpg
n003258_241_4.jpg
n003258_242_4.jpg
n003258_243_4.jpg
n003258_244_4.jpg
n003258_245_4.jpg
n003258_250_4.jpg
n003258_251_4.jpg
n003258_252_4.jpg
n003258_253_4.jpg
n003258_254_4.jpg
n003258_255_4.jpg
n003258_256_4.jpg
n003258_261_4.jpg
n003258_262_4.jpg
n003258_263_4.jpg
n003258_264_4.jpg
n003258_265_4.jpg
n003258_266_4.jpg
n003258_267_4.jpg