

# 電腦視覺與深度學習

## (Computer Vision and Deep Learning)

### Homework 1

TA:

Mark : [29A7882zz@gmail.com](mailto:29A7882zz@gmail.com)

Office Hour: 09:00~11:00, Wed.

19:00~21:00, Mon.

At CSIE 9F Robotics Lab.

# Notices (1/2)

- ❑ Copying homework is strictly prohibited!! **Penalty: Grade will be zero for both persons!!**
- ❑ If the code can't run, you can come to our Lab within one week and show that your programming can work. Otherwise you will get zero!!
- ❑ Due date => **2019/11/06 (Wed.) 23:59:59**
  - No delay. If you submit homework after deadline, you will get 0.
- ❑ Upload to => **140.116.154.1 -> Upload/Homework/HW1**
  - **User ID: cvdl2019      Password: cvdl2019**
- ❑ Format
  - Filename: Hw1\_StudentID\_Name\_Version.rar
    - Ex: Hw1\_F71234567\_林小明\_v1.rar
    - If you want to update your file, you should update your version to be v2, ex: Hw1\_F71234567\_林小明\_v2.rar
  - Content: **project folder**\*( including the pictures )
    - \*note: remove your "Debug" folder to reduce file size

# Notices (2/2)

## ❑ C++ (check MFC guide in ftp)

- OpenCV 3.3.1 (<https://opencv.org/release.html>)
- Visual Studio 2015 (download from <http://www.cc.ncku.edu.tw/download/>)
- UI framework: MFC

## ❑ Python

- Python 3.7 (<https://www.python.org/downloads/>)
- Tensorflow 2.0 / PyTorch 1.3.0
- opencv-contrib-python (3.4.2.17)
- Matplotlib 3.1.1
- UI framework: pyqt5 (5.11.3)

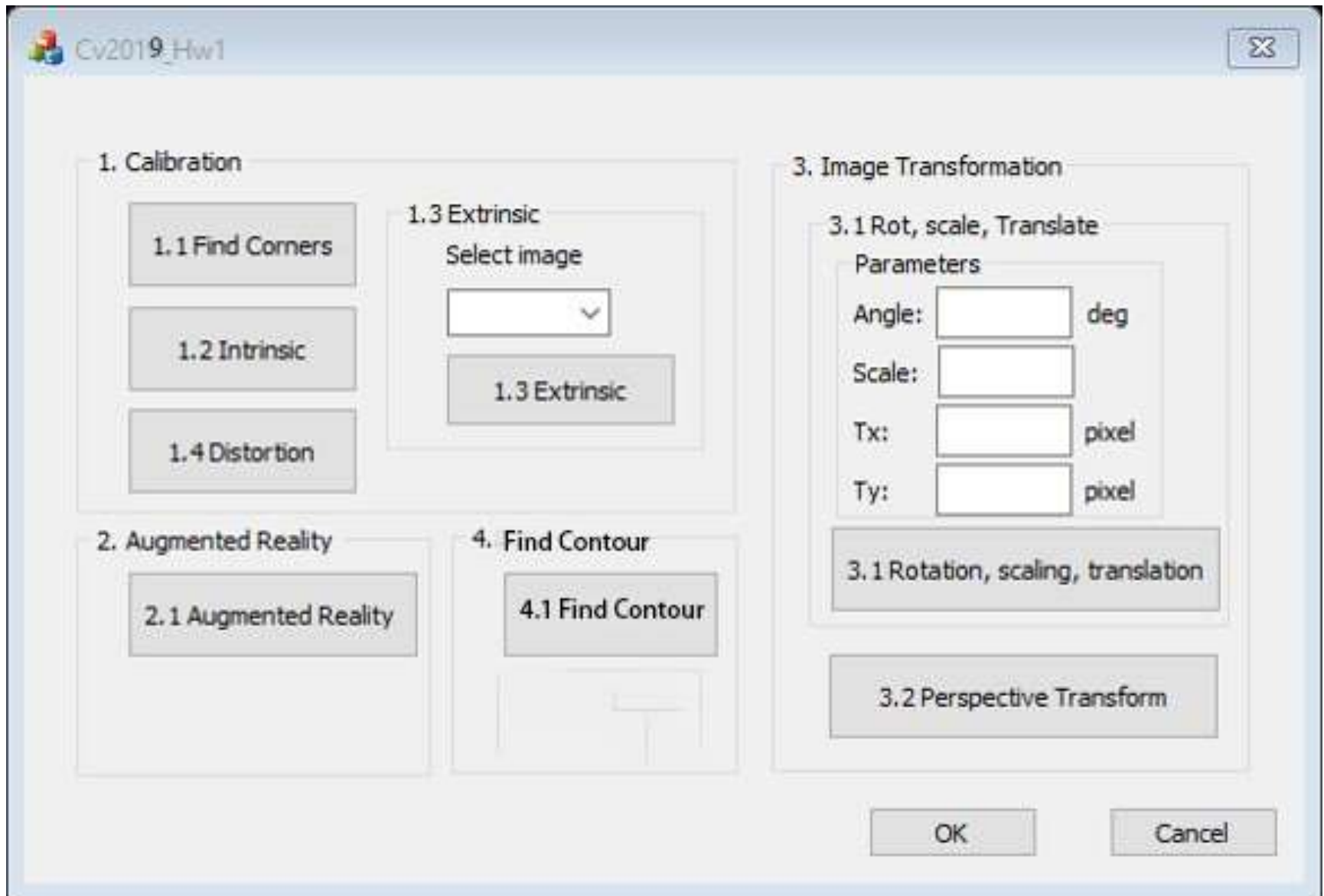
# Grading

## 0. Homework Format

1. (10%, reference) Camera Calibration (出題 : Jang)
  - 1.1 Corner detection (0%)
  - 1.2 Find the intrinsic matrix (0%)
  - 1.3 Find the extrinsic matrix (0%)
  - 1.4 Find the distortion matrix (0%)
2. (20%) Augmented Reality (出題 : Jang)
3. (10%, reference) Image Transformation (出題 : YiYuan)
  - 3.1 Rotation, scaling, translation (0%)
  - 3.2 Perspective transform (0%)
4. (10%, reference) Find Contour (出題 : Rex)
5. (50%) Training Cifar-10 classifier using LeNet (出題 : Michael)

# 0. Homework Format

- ❑ Use MFC to create GUI like following picture



# 1. Camera Calibration

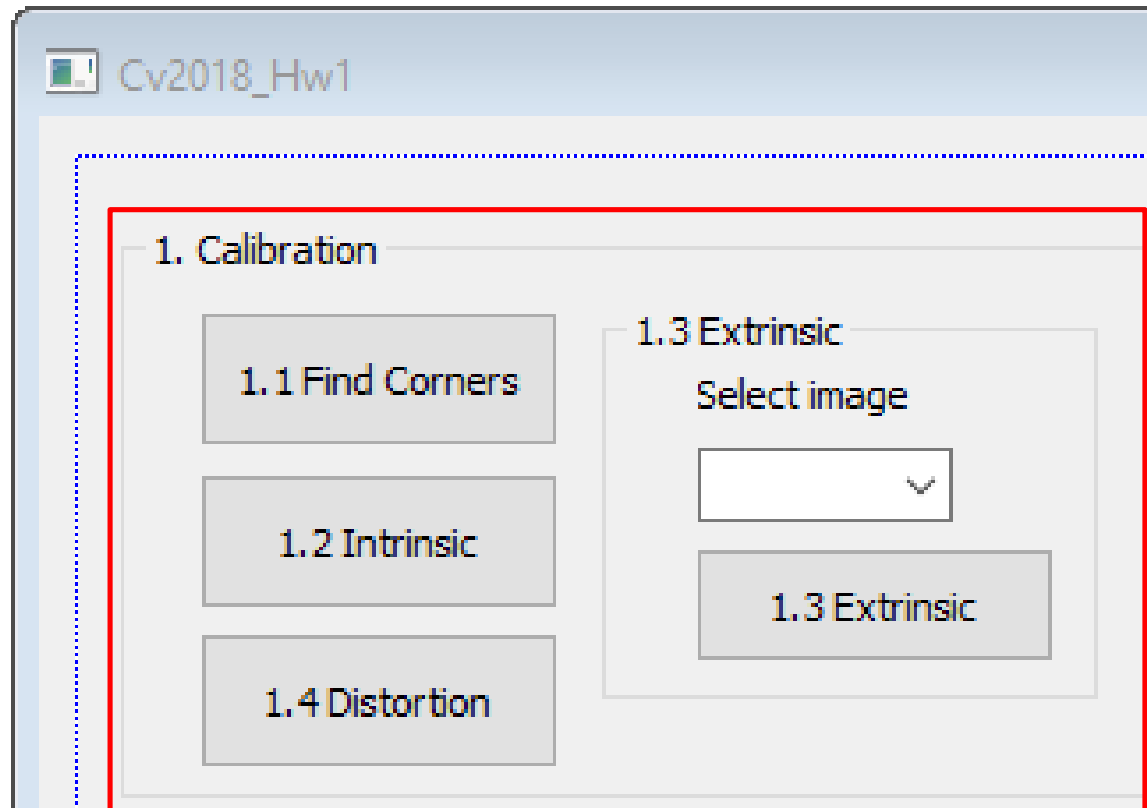
(出題 : Jang)

1.1 Corner detection

1.2 Find the intrinsic matrix

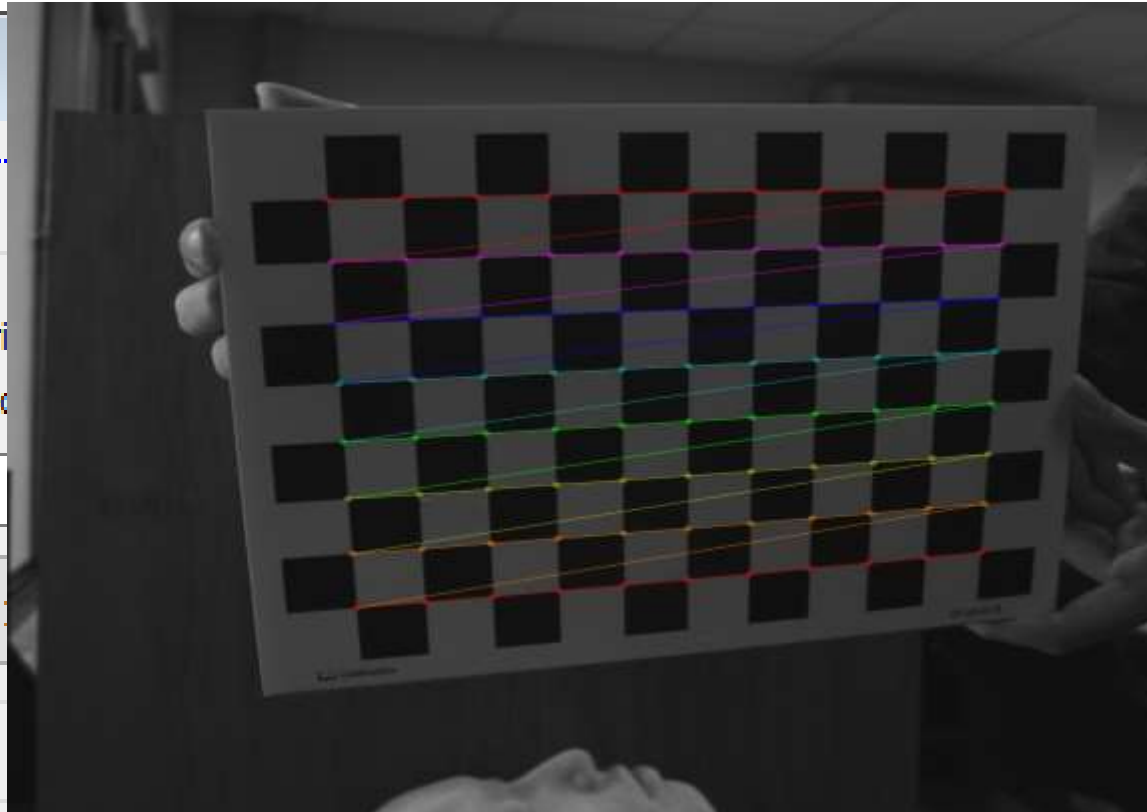
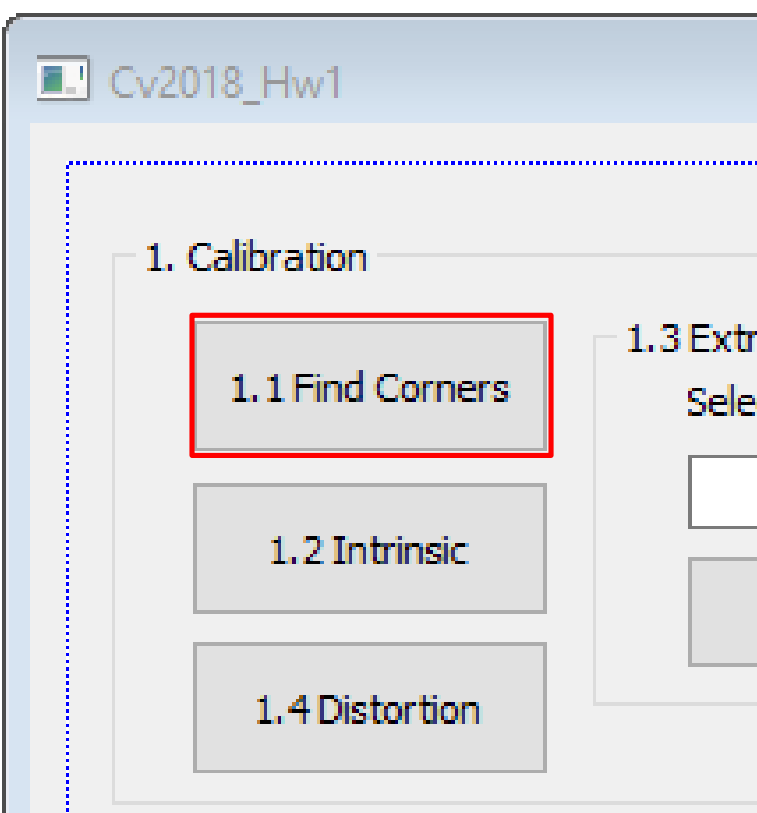
1.3 Find the extrinsic matrix

1.4 Find the distortion matrix



# 1.1 Corner Detection

- ❑ Given: 15 images, 1.bmp ~ 15.bmp
- ❑ Q: 1) Find and draw the corners on the chessboard for each image.  
2) Click button “1.1” to show the result.
- ❑ Hint :  
OpenCV Textbook Chapter 11 (p. 398 ~ p. 399)  
`cvShowImage(...);`
- ❑ Ex:



# 1.2 Find the Intrinsic Matrix

❑ Given: 15 images, 1.bmp ~ 15.bmp

❑ Q: 1) Find the intrinsic matrix ():

$$\begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

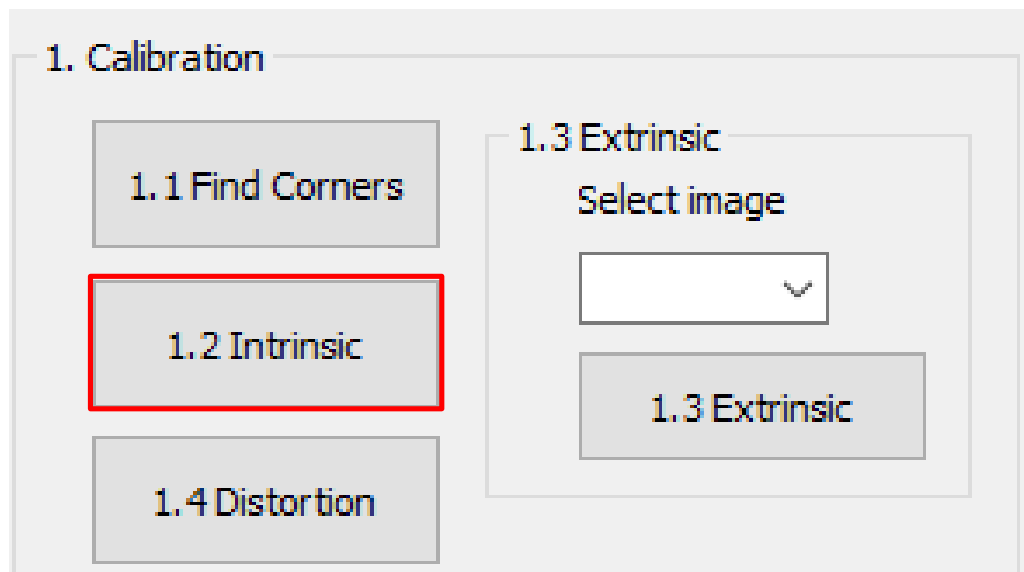
2) Click button “1.2” and then show the result on the console window.

❑ Output format:

```
[2227.333008, 0.000000, 384.186066;  
0.000000, 2226.654541, 299.351746;  
0.000000, 0.000000, 1.000000]
```

(Just an example)

❑ Hint: OpenCV Textbook Chapter 11 (P.398 ~ p.400)





# 1.3 Find the Extrinsic Matrix

- Given: intrinsic parameters, distortion coefficients, and the list of 15 images
- Q: 1) Find the extrinsic matrix of the chessboard for each of the 15 images, respectively:

$$\begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \end{bmatrix}$$

2) Click button “1.3” and then show the result on the console window.

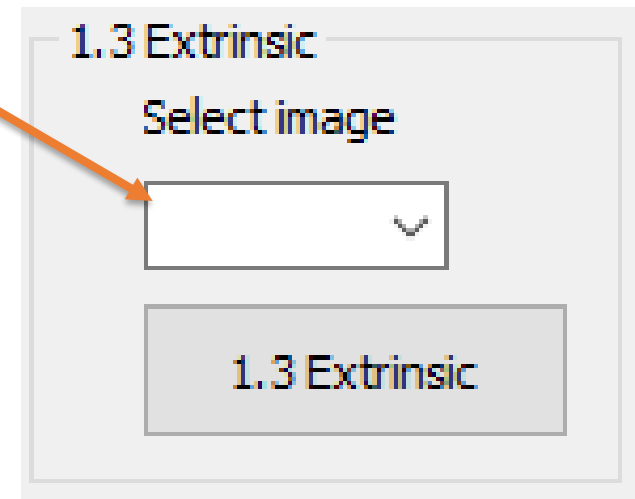
- Output format:

```
[ -0.128827 , 0.991169 , -0.031426 , -1.969988 ;  
 0.983549 , 0.131755 , 0.123583 , -1.105037 ;  
 0.126632 , -0.014988 , -0.991836 , 49.121323 ; ]
```

(Just an example)

- Hint: OpenCV Textbook Chapter 11, p.370~402

- (1) List of numbers: 1~15
- (2) Select 1, then 1.bmp will be applied, and so on



# 1.4 Find the Distortion Matrix

❑ Given: 15 images

❑ Q: 1) Find the distortion matrix:  $[k_1, k_2, p_1, p_2, k_3]$

2) Click button “1.4” to show the result on the console window.

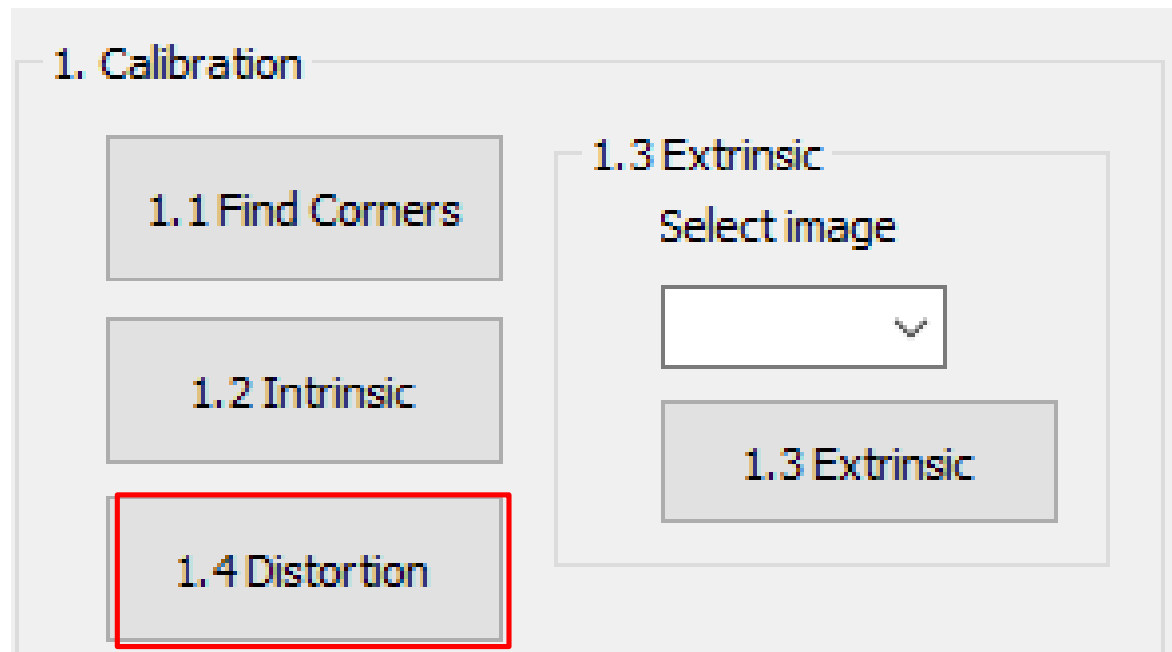
❑ Output format:

```
[-0.072230, -0.261944, -0.000024, -0.003354, 4.228090]
```

(Just  
an example)

❑ Hint:

- Distortion coefficients can be obtained simultaneously with intrinsic parameters
- OpenCV Textbook Chapter 11 (P.398 ~ p.400)



## 2. Augmented Reality

(出題 : Jang)

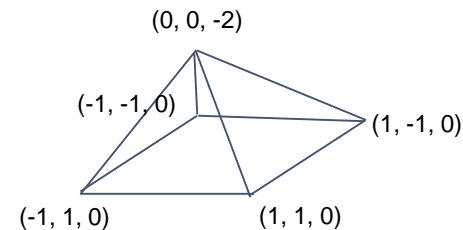
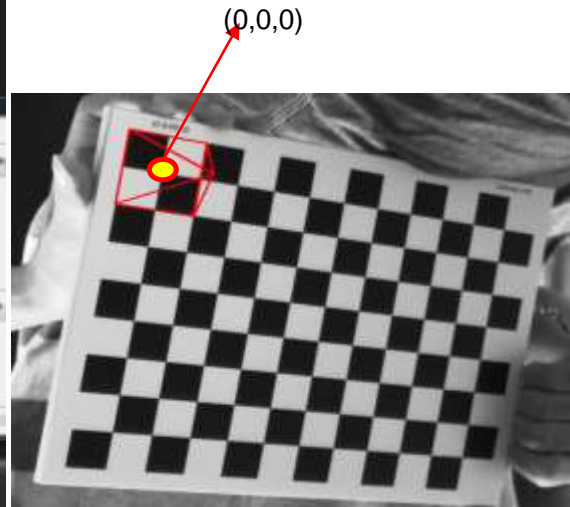
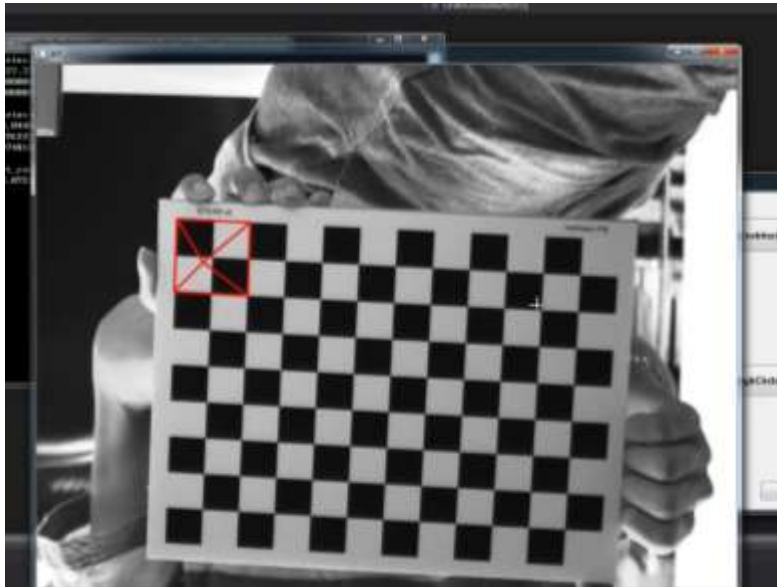
- ❑ Given: Intrinsic and extrinsic parameters, distortion coefficients, and 5 images: 1~5.bmp
- ❑ Q: 1) Draw a “pyramid” on the chessboard images (1.bmp to 5.bmp), then  
2) Click the button ”2” to show the pyramid on the picture. Show each picture for 0.5 seconds (total 5 images)
- ❑ Hint : Textbook Chapter 11, p.370~402

• 3D Object coordinates:

Vertex  $(0, 0, -2)$

Corners  $(1, 1, 0)(1, -1, 0)(-1, -1, 0)(-1, 1, 0)$

Demo video:



# 3.1 Transforms: Rotation, Scaling, Translation

(出題 : YiYuan) ??

❑ Given: **OriginalTransform.png** image

❑ Q: 1) Click button “3.1”, **OriginalTransform.png** should be showed.

2) Please rotate, scale and translate the **small squared image** (as Figure 3.1) with following parameters (**should be entered in the GUI**):

(1) **Angle = 45°** (counter-clockwise)

(2) **Scale = 0.8**,

(3) **Translation with:**

$$\square x_{\text{new}} = x_{\text{old}} + 150 \text{ pixels} = 130 + 150 = 280$$

$$\square y_{\text{new}} = y_{\text{old}} + 50 \text{ pixels} = 125 + 50 = 175$$

**Point C (130,125) is center of small square image**

❑ Hint :

OpenCV Textbook Chapter 12 (p. 407 ~ p. 412)

`warpAffine(...);`

❑ EX:

Small square image

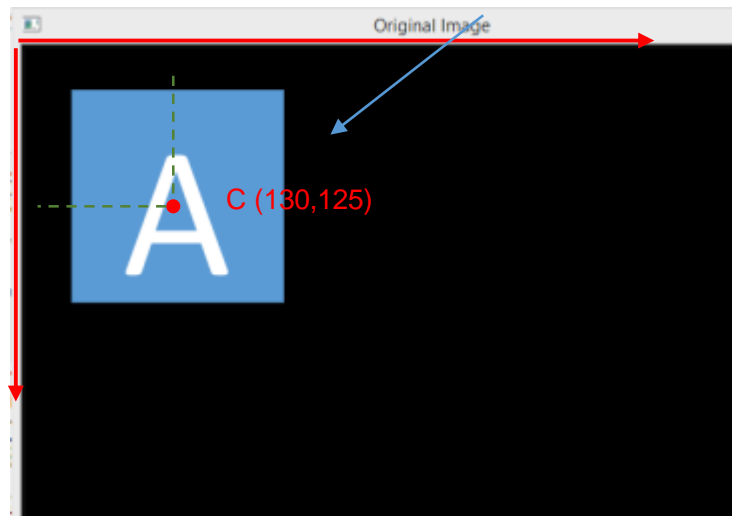
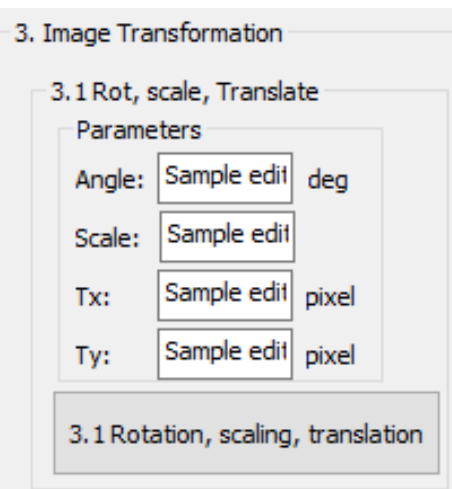


Figure 3.1 Original Image

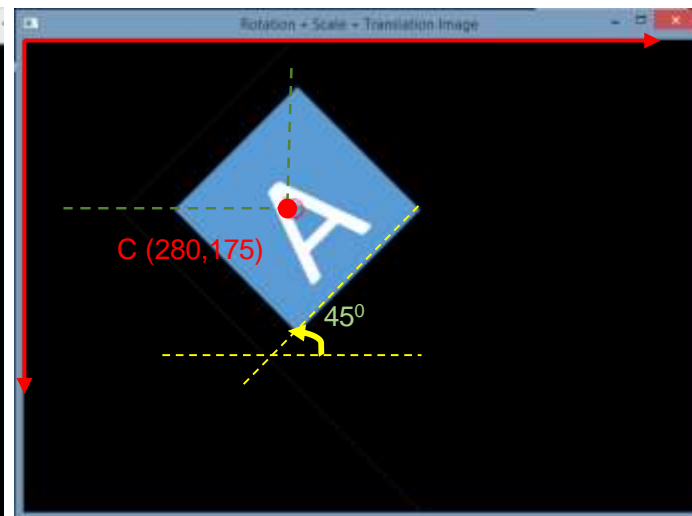
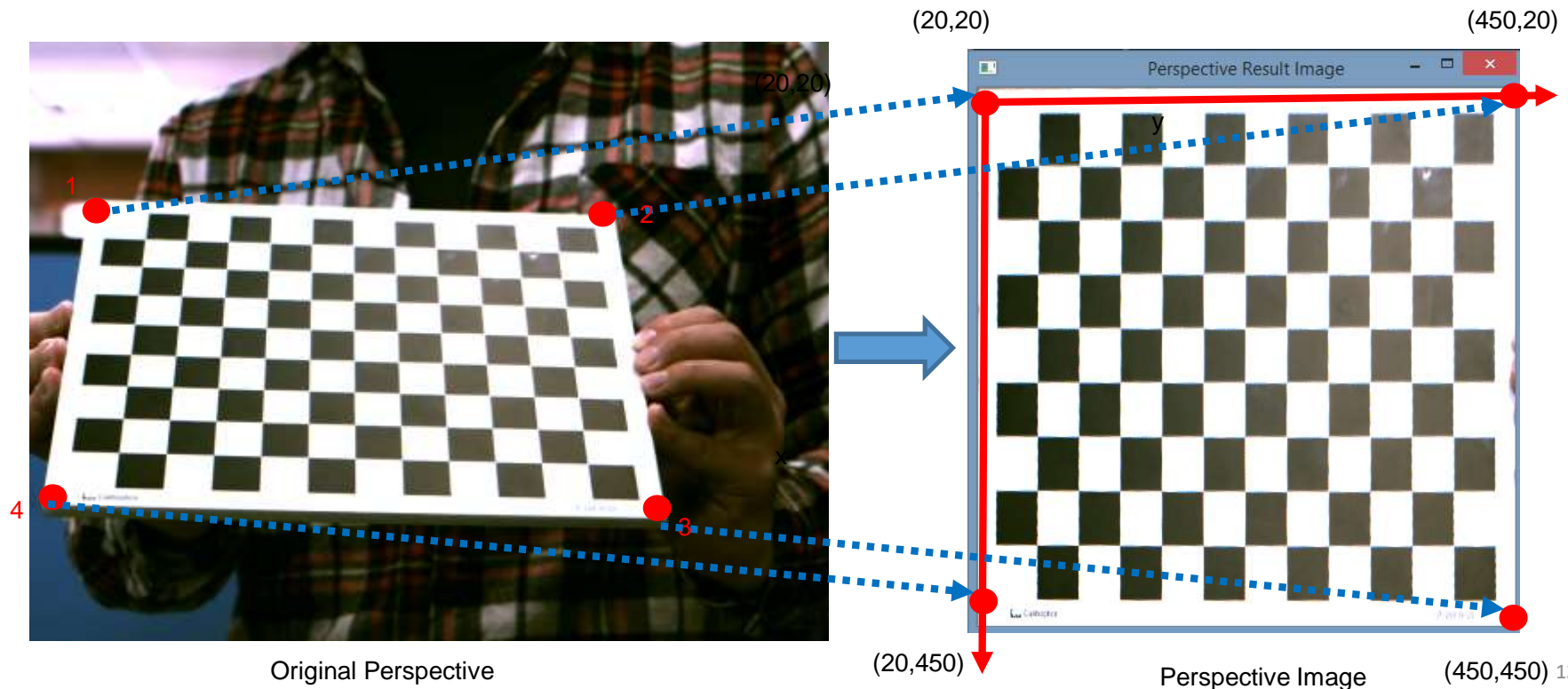


Figure 3.2 Rotation, Scale and Translation Image

## 3.2 Perspective Transformation

(出題 : YiYuan)

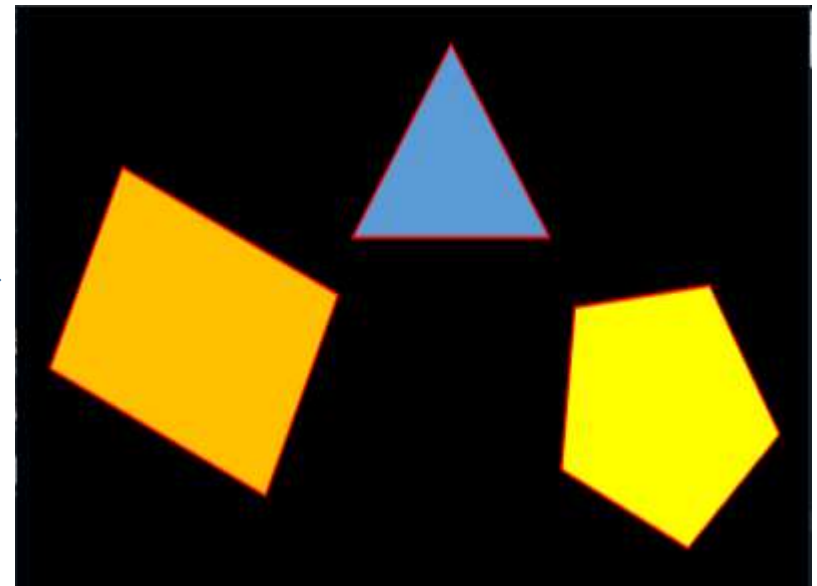
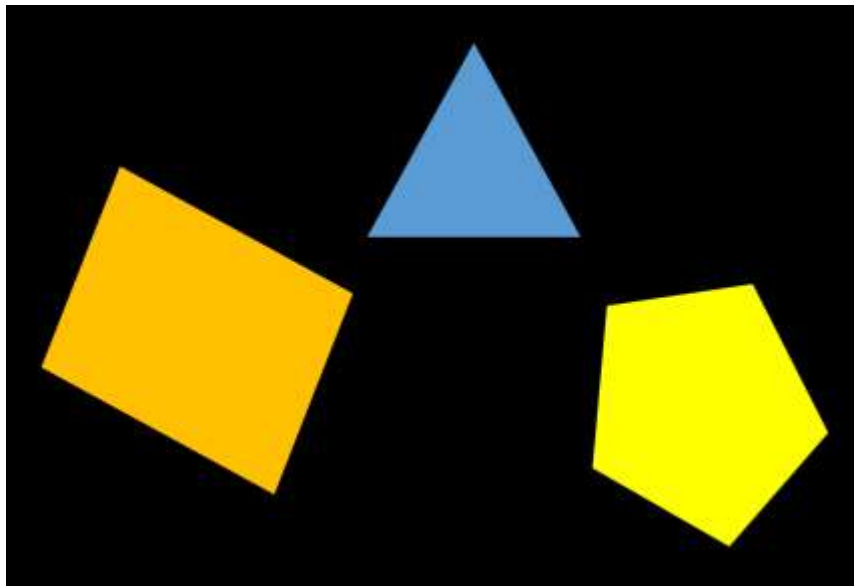
- ❑ Given: **OriginalPerspective.png** image
- ❑ Q: Use OpenCV functions to project the image
  - 1) Click button “3.2” to show image in the new window. Then do:
    - a) Click 4 points showed in console window. (start from top-left corner of the original image, and then click clock-wise)
    - b) Warp the original image to the location (20,20), (20,450), (450,450), (450,20). Open second window to show the result.
- ❑ Hint:
  - Textbook Chapter 6, p. 170~171
  - mouse callback function (p.96)



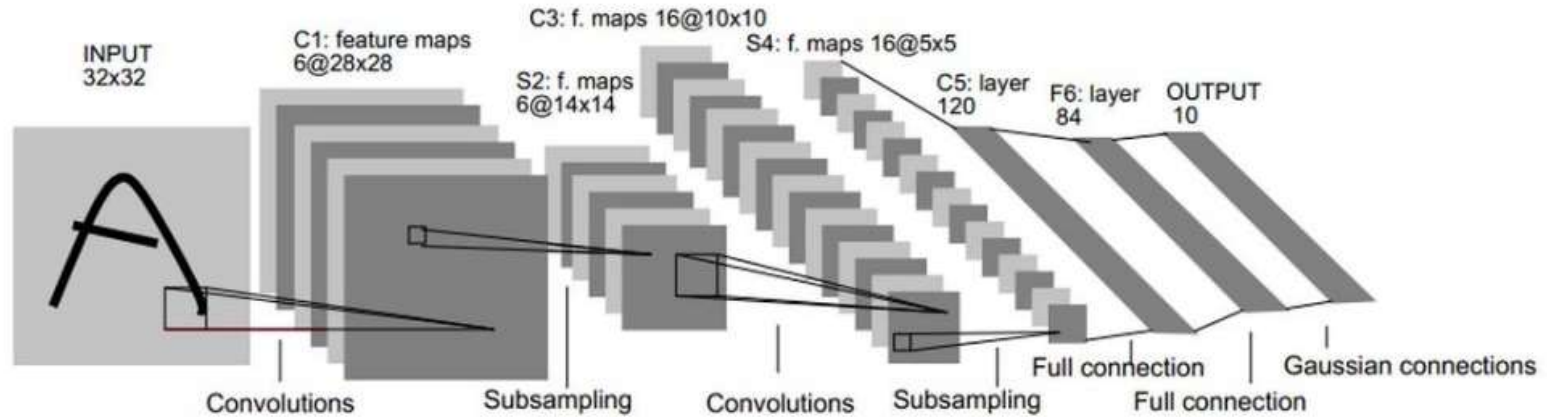
## 4.0 Find Contour

(出題 : Rex)

- ❑ Given: ***Contour.png*** image
- ❑ Q: Use OpenCV functions to find all the contours in the image
  - 1) Click button “4.1” to show image in the new window. Then draw the edges by red.
- ❑ Hint:
  - Textbook Chapter 8, p. 234~241



## 5.0 Train Cifar-10 Classifier Using LeNet-5 (出題 : Michael)



1. Learn to construct LeNet-5 and train it by using data Cifar-10.

2. Environment Requirement

- 1) Python 3.7
- 2) Tensorflow 2.0 / PyTorch 1.3.0
- 3) opencv-contrib-python 3.4.2.17
- 4) Matplotlib 3.1.1

3. Reference

- 1) Gradient-Based Learning Applied to Document Recognition  
(<http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf>) (LeNet)
- 2) Cifar-10 (<https://www.cs.toronto.edu/~kriz/cifar.html>)

5.1 Show Train Images
5.2 Show Hyperparameters
5.3 Train 1 Epoch
5.4 Show Training Result
Test Image Index: <input type="text"/> (0~9999)
5.5 Inference



5.1 Load Cifar-10 training dataset and randomly show 10 images and labels respectively. (10%)

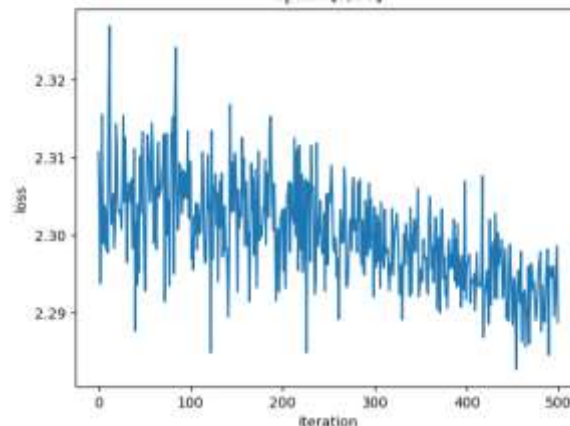


Label:            cat                            ship                            ship                            airplane                    ...

5.2 Print out training hyperparameters (batch size, learning rate, optimizer). (10%)

```
hyperparameters:  
batch size: 32  
learning rate: 0.001  
optimizer: SGD
```

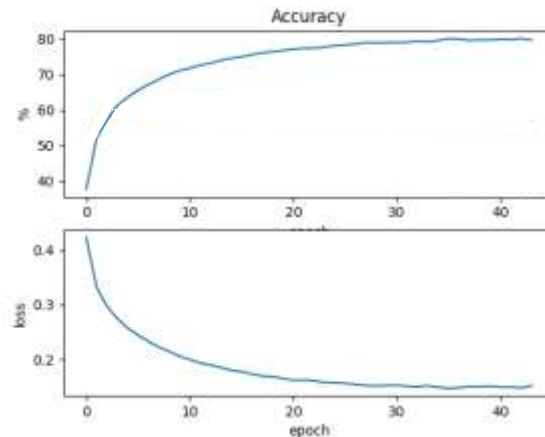
5.3 Train 1 epoch from initial status and show training loss at the end of the epoch. (10%)



(Record loss per iteration)



5.4 Train your model at least 50 epochs **by your own computer**, save your model and take a screenshot of your training loss and accuracy. (10%)



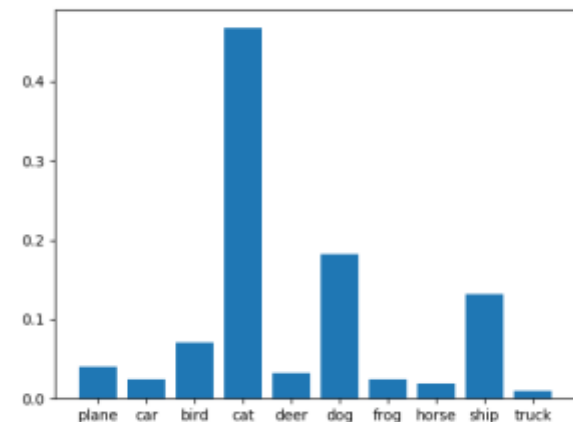
(Record accuracy/loss per epoch)

5.5 Load your model trained at 5.4. Let us choose one test image from Cifar-10 test images. Then inference the image, show image and estimate this test image. (10%)

Test Image Index:



Show image



Estimation result