Deep Learning for Creative Design (CD2025)
设计方法与开发程序(中韩合作) 本科三年级专业课
（11 月 4,11,18,25 日 Tuesday）（产 818）13:20~14:40、15:00~16:20
（11 月 5,12,19,26 日 Wednesday）（产 813）15:00~16:20、16:35~17:55

Deep learning is transforming creative design by enabling new forms of co-creation and exploration in art, music, architecture, fashion, and product design, offering more than just automation—it amplifies human creativity and experimentation.

Design + AI convergence. Learning generative design and data-driven design. Focused on practical projects, studio classes, and industry-academia collaboration. Aiming to cultivate both “designer sensibility and technical skills.”

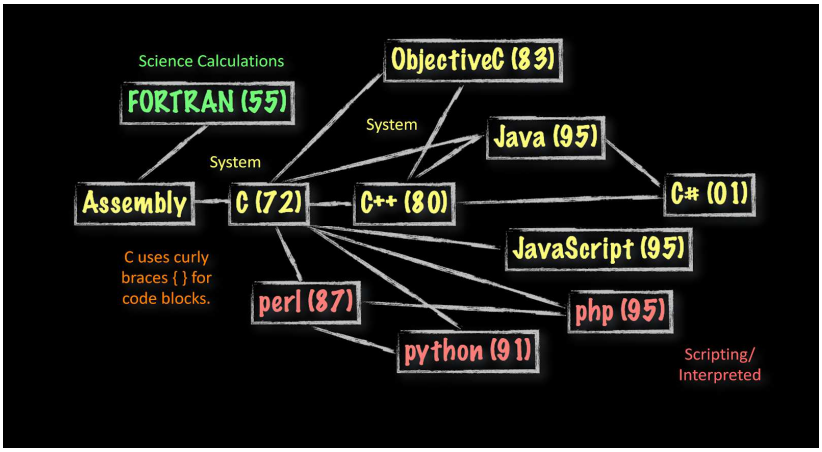
- Generative Models: Tools like GANs (e.g., StyleGAN) and diffusion models (e.g., Stable Diffusion, DALL·E) can create original images, textures, and patterns.
- Style Transfer: Transform an image into the style of a famous painting (Van Gogh, Picasso) while keeping its content.
- Logo & Layout Generation: Neural networks can suggest variations of logos, web layouts, and branding materials.
- **Generative AI + Fine Tuning ###**

Text and Reference
Python Tutorial Python Software Foundation ###
OpenCV Python Tutorial Alexander Mordvintsev & Abid K
GenAIFineTuning ###

[11.4,5]
Attendance Check
GenAIFineTuning ###
Homework1 Registration

Python, Python Tutorial ###
online-python.com programiz
Chap 3 : An Informal Introduction to Python ###
Chap 4 : More Control Flow Tools ###
Chap 5 : Data Structures ###
Chap 6 : Modules ###
Chap 7 : Input and Output
Chap 8 : Errors and Exceptions
Chap 9 : Classes ###

https://en.wikipedia.org/wiki/History_of_programming_languages



source : PHP-01-Overview.ppt@wa4e

https://code.visualstudio.com/
Anaconda | The World's Most Popular Data Science Platform

```
conda update conda
conda create -n CD2025 python=3.8 anaconda
conda activate CD2025
pip install --upgrade pip
pip install opencv-python
pip install matplotlib

conda
conda -V
conda update conda
conda create -n yourenvname python=3.# anaconda

conda activate yourenvname
conda deactivate
conda env list
conda remove -n yourenvname --all

pip list
pip install --upgrade tensorflow
pip install pygame
pip uninstall matplotlib
pip install -r requirements.txt
```

https://colorscripter.com/


leebyunggook
대한민국 부산

https://pythonbasics.org/exercises/
https://www.w3schools.com/python/default.asp
An introduction to Python with Numpy, Scipy, and Matplotlib
Object Oriented Design with Python, Classes, Objects and Class
https://edabit.com/challenges/python3

GitHub URL Registration (github, gitee)
github Markdown README.md
github readme.md markdown 8:37
https://github.com/intel-iot-devkit/sample-videos

OpenCV Python Tutorial, Alexander Mordvintsev & Abid K

chap1.2 GUI Features in OpenCV p.19
1.2.1 Getting Started with Images p.21
Show Image with openCV

```
import numpy as np
import cv2
img = cv2.imread('mountain.jpg',0)
cv2.imshow('image',img)
k = cv2.waitKey(0)
if k == 27:                          # wait for ESC key to exit
  cv2.destroyAllWindows()
elif k == ord('s'):                  # wait for 's' key to save and exit
  cv2.imwrite('mountain_gray.png',img)
cv2.destroyAllWindows()
```

Show Image with pyplot

```
import numpy as np
import cv2
from matplotlib import pyplot as plt
img = cv2.imread('mountain.jpg',0)
```

```
plt.imshow(img, cmap = 'gray', interpolation = 'bicubic')
plt.xticks([]), plt.yticks([])                    # to hide tick values on X and Y axis
plt.show()
```

1.2.2 Getting Started with Videos p.24

Capture Video from Camera

```
import numpy as np
import cv2
cap = cv2.VideoCapture(0)
while(True):                                      # Capture frame-by-frame
 ret, frame = cap.read()
   gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # Our operations on the frame
 cv2.imshow('frame',gray)                         # Display the resulting frame
 if cv2.waitKey(1) & 0xFF == ord('q'):
   break
cap.release()                                     # When everything done, release the capture
cv2.destroyAllWindows()
```

Playing Video from file

```
cap = cv2.VideoCapture('vtest.avi')
while(cap.isOpened()):
```

Saving a Video

```
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('output.avi',fourcc, 20.0, (640,480))
while(cap.isOpened()):
 ret, frame = cap.read()
 if ret==True:
   frame = cv2.flip(frame,0)
   out.write(frame)                               # write the flipped frame
```

1.2.3 Drawing Functions in OpenCV p.27

```
import numpy as np
import cv2
img = np.zeros((512,512,3), np.uint8)             # Create a black image
img = cv2.line(img,(0,0),(511,511),(255,0,0),5)   # Draw a diagonal blue line
img = cv2.rectangle(img,(384,0),(510,128),(0,255,0),3)
img = cv2.circle(img,(447,63), 63, (0,0,255), -1)
img = cv2.ellipse(img,(256,256),(100,50),0,0,180,255,-1)
pts = np.array([[10,5],[20,30],[70,20],[50,10]], np.int32)
pts = pts.reshape((-1,1,2))
img = cv2.polylines(img,[pts],True,(0,255,255))
font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(img,'OpenCV',(10,500), font, 4,(255,255,255),2,cv2.LINE_AA)
```

1.2.4 Mouse as a Paint-Brush p.30

```
import cv2
import numpy as np
drawing = False                                   # true if mouse is pressed
mode = True                                       # if True, draw rectangle. Press 'm' to toggle to curve
ix,iy = -1,-1

def draw_circle(event,x,y,flags,param):
 global ix,iy,drawing,mode
 if event == cv2.EVENT_LBUTTONDOWN:
   drawing = True
   ix,iy = x,y
 elif event == cv2.EVENT_MOUSEMOVE:
   if drawing == True:
     if mode == True:
       cv2.rectangle(img,(ix,iy),(x,y),(0,255,0),-1)
     else:
       cv2.circle(img,(x,y),5,(0,0,255),-1)
 elif event == cv2.EVENT_LBUTTONUP:
   drawing = False
```

```
    if mode == True:
      cv2.rectangle(img,(ix,iy),(x,y),(0,255,0),-1)
    else:
      cv2.circle(img,(x,y),5,(0,0,255),-1)
img = np.zeros((512,512,3), np.uint8)
cv2.namedWindow('image')
cv2.setMouseCallback('image',draw_circle)                    # mouse callback function

while(1):
  cv2.imshow('image',img)
  k = cv2.waitKey(1) & 0xFF
  if k == ord('m'):
    mode = not mode
  elif k == 27:
    break
cv2.destroyAllWindows()
```

1.2.5 Trackbar as the Color Palette p.32

```
import cv2
import numpy as np

def nothing(x):
  pass

img = np.zeros((300,512,3), np.uint8)                        # Create a black image, a window
cv2.namedWindow('image')
cv2.createTrackbar('R','image',0,255,nothing)                # create trackbars for color change
cv2.createTrackbar('G','image',0,255,nothing)
cv2.createTrackbar('B','image',0,255,nothing)
switch = '0 : OFF \n1 : ON'
cv2.createTrackbar(switch, 'image',0,1,nothing)              # create switch for ON/OFF functionality

while(1):
  cv2.imshow('image',img)
  k = cv2.waitKey(1) & 0xFF
  if k == 27:
    break
  r = cv2.getTrackbarPos('R','image')                        # get current positions of four trackbars
  g = cv2.getTrackbarPos('G','image')
  b = cv2.getTrackbarPos('B','image')
  s = cv2.getTrackbarPos(switch,'image')
  if s == 0:
    img[:] = 0
  else:
    img[:] = [b,g,r]
cv2.destroyAllWindows()
```

MediaPipe mediapipe.zip ###
https://github.com/google-ai-edge/mediapipe
mediapipe 19:07 faceSwap 0:47 MultiPose2Single 0:26
mediapipe0 25:42 mediapipe1 28:56 mediapipe2 25:40

Multi-person Real-time Action Recognition Based-on Human Skeleton ###
humanActionRecog0 25:56 humanActionRecog1 26:09
https://www.youtube.com/watch?v=h8nu1Sts1sU 1:30
https://www.youtube.com/watch?v=abpZQM20608 1:28

chap1.4 Image Processing in OpenCV
1.4.3 Geometric Transformations of Images p.58

1.4.4 Smoothing Images p.62
Convolution https://youtu.be/mQDHAn1DrHs
https://en.wikipedia.org/wiki/Kernel_(image_processing)

https://en.wikipedia.org/wiki/Convolution
https://en.wikipedia.org/wiki/Lenna
chap144kc.ipynb lena.png 이미지 bilateral filtering 구현하기
https://en.wikipedia.org/wiki/Median_filter
cv2.filter2D | LearnOpenCV

1.4.5 Morphological Transformations p.68
tut1.ppt by 虞台文博士, Tatung University, Taiwan
MorphologicalOperator
FingerPrint data set Morphological Operator

Dilation 확장 $A \oplus B = \{z|(B)_z \cap A \neq \emptyset\}$

Erosion 침식 $A \ominus B = \{z|(B)_z \subseteq A\}$

Opening 열림 $A \bigcirc B = (A \ominus B) \oplus B$

Closing 닫힘 $A \bullet B = (A \oplus B) \ominus B$

1.4.6 Image Gradients
1.4.7 Canny Edge Detection
EdgeDetect.pdf Canny Edge Detection
레나 (이미지) - 위키백과, 우리 모두의 백과사전
CS589NM Digital Image Processing - Lect 1. Introduction & Fundamentals (pdf)

1.4.8 Image Pyramids **(skip)**
1.4.9 Contours in OpenCV (p.85~94)
Contours Data Set ch149.ipynb ←code
Count Coin : data set with canny edge detection & contour analysis (with your data set)

Lecture 2. Intensity Transformation and Spatial Filtering (pdf)
1.4.10 Histograms in OpenCV (p.106~118)
Histogram Data Set

Lecture 3. Filtering in the Frequency Domain (pdf)
1.4.11 Image Transforms in OpenCV **(skip)**
1.4.12 Template Matching (Data Set)
1.4.13 Hough Line Transform Hough Transformation (ppt)
1.4.14 Hough Circle Transform

chap1.5 Feature Detection and Description
1.5.1 Understanding Features **(skip)**
1.5.2 Harris Corner Detection (ppt)

https://mattabrown.github.io/autostitch.html
https://www.microsoft.com/en-us/research/project/image-composite-editor/
https://en.softonic.com/download/microsoft-image-composite-editor-64bits/windows/post-download

1.5.3 Shi-Tomasi Corner Detector & Good Features to Track
1.5.4 Introduction to SIFT(Scale-Invariant Feature Transform)

pip uninstall opencv-python
pip install opencv-contrib-python

youTube 1 2 3
http://vision.middlebury.edu/stereo/data/
http://www.vlfeat.org/overview/sift.html

1.5.5 Introduction to SURF (Speeded-Up Robust Features) **(skip)**
1.5.6 FAST Algorithm for Corner Detection **(skip)**
1.5.7 BRIEF (Binary Robust Independent Elementary Features) **(skip)**

1.5.8 ORB (Oriented FAST and Rotated BRIEF) **(skip)**

1.5.10 Feature Matching + Homography to find Objects
https://pythonhosted.org/sift_pyocl/sift.htm
http://www.cs.cornell.edu/courses/cs4670/2015sp/projects/pa2/
https://www.kaggle.com/wesamelshamy/tutorial-image-feature-extraction-and-matching
https://www.pyimagesearch.com/2015/07/16/where-did-sift-and-surf-go-in-opencv-3/
https://pypi.org/project/opencv-contrib-python/

implement SIFT feature matching with camera.
Feature Matching (Brute-Force) – OpenCV 3.4 with python 3 Tutorial 26 - Pysource
matchHomography : data set

1.6.1 Meanshift and Camshift reference chap161.ipynb
1.6.2 Optical Flow reference reference chap162.ipynb
1.6.3 Background Subtraction reference chap163.ipynb
https://github.com/intel-iot-devkit/sample-videos

chap1.6 Video Analysis
chap1.7 Camera Calibration and 3D Reconstruction
1.7.1 Camera Calibration Image.zip chap171.ipynb
AR with Python & OpenCV Calibration
1.7.2 Pose Estimation chap172.ipynb

1.7.3 Epipolar Geometry chap173.ipynb Multiple View Geometry
1.7.4 Depth Map from Stereo Images chap174.ipynb
https://vision.middlebury.edu/stereo/data/scenes2014/datasets/Playtable-perfect/
CodedStructureLight_bglee1911.pdf