

CMPE258
Fall 2023

1/

August 22 (Tue)

Organizational meeting.

1. Class material on github
github/hualili

alv100	Add files via upload
deep-learning-2020s	Add files via upload
deep-learning-2022s	Add files via upload
deep-learning-2023s	Add files via upload
facial-detect	Add files via upload
lec1 Capture/CMakeFiles	Delete CMakeDirectory/Information...
lecOpenCV_GL	openGL and openCV sample commi...
riscv	Create readme.txt
20-2021S-0-7-1convnets-NumeraiD...	Add files via upload

Course and Contact Information

Instructor(s): Harry Li

Office Location: Engineering Building, Room 267A

Telephone: (650) 400-1116 for text messaging only

Email: hua.li@sjsu.edu

Office Hours: M.W. 3:00-4:00 pm

In-Person.

Class Days/Time: Tuesdays and Thursdays 4:30-5:45 pm.

Classroom: Engineering Building Room 337

Prerequisites: CMPE 255 or CMPE 257 or instructor consent. Computer Engineering majors only.

Course Description

2. Prerequisites Requirements

Bring your Proof to the next Class.

3. Emphasis on "Deep Neural Networks", &
Semantic Segmentation

Course Description

Deep neural networks and their applications to various problems, e.g., speech recognition, image segmentation, detection and recognition of temporal and spatial patterns, and natural language processing. Covers underlying theory, the range of applications to which it has been applied, and learning from very large data sets.

Note: Definition (HL): (Human Intelligence)
is Symbolic Representation of
Learned Experience.



4. Projects. 2

plws 1 team project
(Semester Long) } ~~30%~~ 25%
30pts

5. In-Person Class; CANVAS is

utilized to post Homework / Project Requirements, and to collect the submission of the homework, as well as for the exams.

This course is an online course. The students must have Internet connectivity and access to their machine. The students must participate in the class activities and submit all assignments, exams to SJSU CANVAS. The syllabus, faculty contact information on the syllabus, projects, and exam papers are all available on CANVAS. See [University Policy F13-2](#)

6.

Grading Information

Quiz, Homework, Projects	30%
Midterm Examination	30%
Final Examination	40%

7. Textbooks & References

Textbook

- Deep Learning with Python, 1st or 2nd Edition, by François Chollet, ISBN-10: 9781617294433, <https://github.com/fchollet/deep-learning-with-python/blob/master/2018F-6-DeepLearningCh02.pdf>
- Robot Vision by B.K. P. Horn, the MIT press, ISBN 0-262-08159-8, (Hill).
- Reference textbook Learning OpenCV, Computer Vision with the OpenCV Library, Kaebler, O'Reilly Publisher, ISBN 978-0-596-51613-0, 2011.

Note:

- 1° CANVAS To Be up by the end of the day, Friday;
- 2° Tools & Software To be installed (Will provide "Readme" as Ref)

OpenCV

Python.

T.F. Version 2.0 or higher

Today's Topic:

Intro. to Deep Convolutional Neural Networks.

Ref: github.

2022F-103-NN-Intro-Python-v5-2022-8-25

Note: Lab Space for the class Rm 268.

8. Software Tools & Dev. Environment

Python. Pycharm.

Anaconda.

TensorFlow.

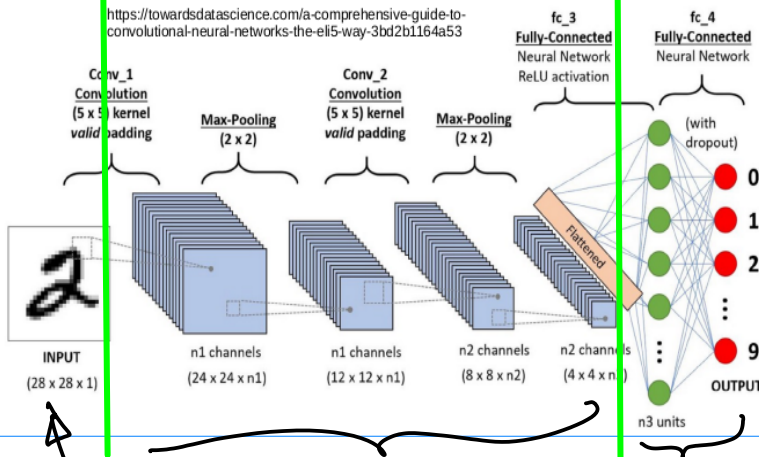
Note: Rm 268 Available per ON
Approved Basis.

August 24 (Thursday)

Example: Architecture Overview.

Note 1.

Illustration of A CNN for Digits Recognition



Preprocessing
Computer
Vision.

Convolution
Layers

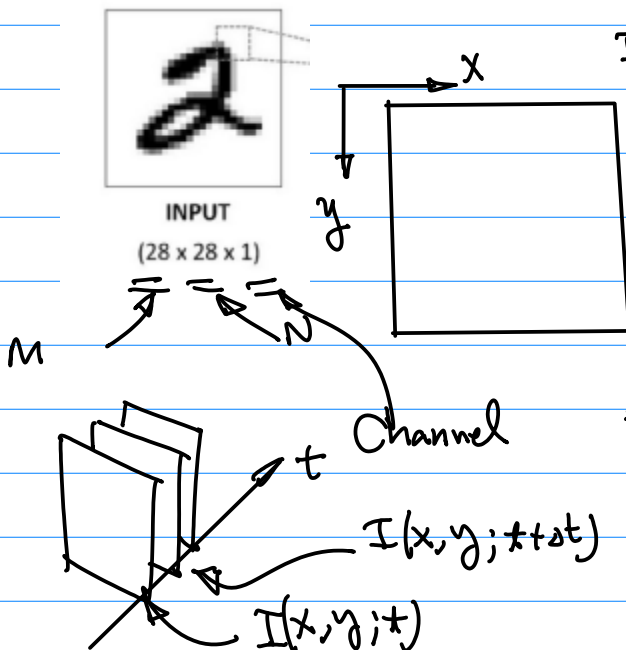
Feature Extraction
To produce Feature Vectors.

$$\vec{u} = (u_1, u_2, \dots, u_n)$$

Feed Forward
Neural Networks.

Decision
Making

2. Image Definition



$$I(x, y; t)$$

Note: for x & y definition
see Ref. Next page.

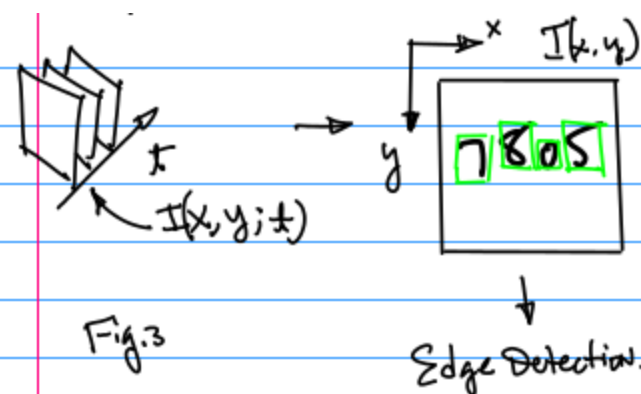
$$M \times N$$

No. of col.
Per Row.

No. of
Rows per frame

Ref: on the github.

2023S-101-Note-cmpe258-2023-03-16.pdf



Comparison to Web Cam.

1080p Resolution: 1920×1080

$$\begin{aligned} \sim 2^{22} &= 2 \cdot 2^{20} \\ &= 2^k \\ \sim 2^{22} &= 2^2 \cdot 2^{20} \\ &= 4 \text{ Meg} \end{aligned}$$

≈ 4 million.

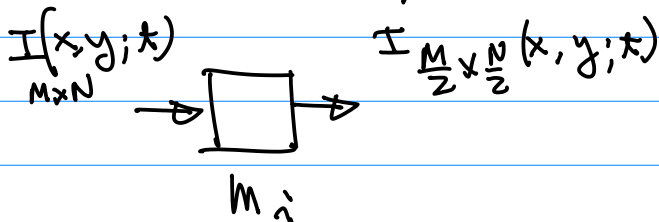
3. For the Convolutional Layer, we denote it as C_i , where $i=1, 2, \dots, N$

for Max Pooling Layer, we denote it as M_j , where $j=1, 2, \dots, K$

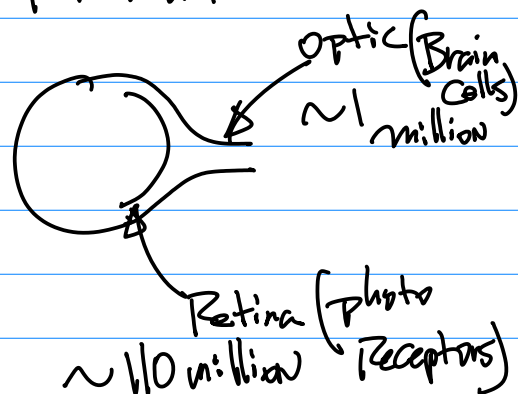
So, for the example, we have

$$C_1 M_1 \rightarrow C_2 M_2$$

Maxpooling for Resolution Reduction / Feature Reduction.



Note: Biologic Inspiration,
Human Visual Interception System,
Retina, ~ 110 million
photo Receptors



4. At the 3rd Segment (Blocks) of the Architecture, we denote Feed Forward Neural Networks as F_M

No. of Neurons / Nodes

for example, F_{10} (10 Nodes) for the output Layer.

August 29 (Tue)

Note: 1° CANVAS is up.

2° Homework Assignment

- a. Honesty pledge to Be Signed / Signed Copy has to be uploaded to CANVAS.

b. Software Tools Installation. (Opt)

- ① OPENCV. By Friday Next Tuesday. Bring your Laptop w/ OPENCV installed.

↓ Please use smartphone to take a photo, And upload the photo to your laptop to display.

Note: Sample code (Python) was posted on the github.

- ② Anaconda Installed on your Laptop.

Note: Readme for Anaconda installation was posted on the github.

- ③ Create ChatGPT Account.

Python Interface to ChatGPT API (3.5 Version) is to be utilized in your Team Project.

3° Form ^ATeam for the Semester Long Project.

Example: Consider 2D Convolution Technique.

Ref: [github/fhualili/opencv](https://github.com/fhualili/opencv)

Note 1:

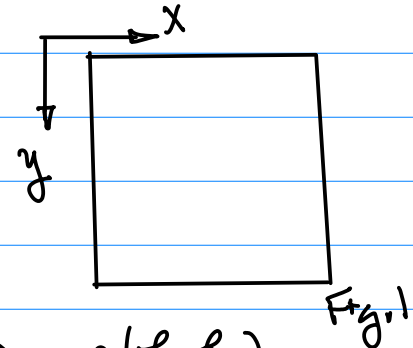
$$c(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} a(k_1, k_2) b(n_1 - k_1, n_2 - k_2)$$

Image Kernel

... (1)

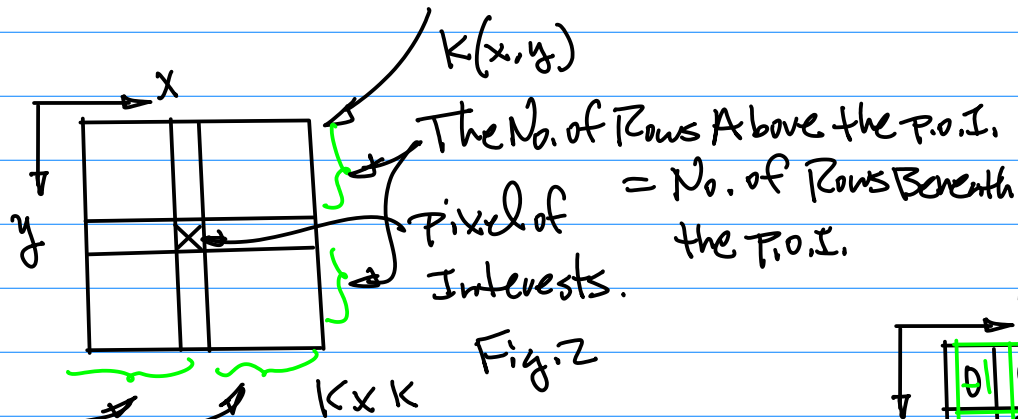
$n_1 = x, n_2 = y$

Summation Index:
 k_1, k_2 are for x and y .



Note 2. $a(x, y)$ or $a(k_1, k_2)$ as an Image. $I(x, y)$
 $b(x, y)$: a Kernel for 2D Convolution, $b(k_1, k_2)$

Note 3. Kernel $b(x, y)$ Can be rewritten $K(x, y)$
Size of A Kernel is denoted as $K \times K$



$K = 3, 5, 7, \dots$ (odd Number)

The No. of Col. Right to the P.O.I. = The No. of Col. Left to the P.O.I.

Fig. 3c

$$\begin{aligned}
 & -1 \times 0 + 0 \times 0 + 1 \times 100 + \\
 & -1 \times 0 + 0 \times 0 + 1 \times 100 + \\
 & -1 \times 0 + 0 \times 0 + 1 \times 100 \\
 & = 300
 \end{aligned}$$

$I(x,y) * K(x,y)$

Fig. 3a

4x4

Fig. 3d

8 bit Grayscale: $2^8 - 1 = 255$

Fig. 3b

Step 2. Shift $K(x,y)$ to the Right By 1 pixel (on the same Row).

Fig. 3e

Step 1. Take the Kernel, and place it at the initial position.

$$\begin{aligned}
 & -1 \times 0 + 0 \times 100 + 1 \times 100 + \\
 & -1 \times 0 + 0 \times 100 + 1 \times 100 + \\
 & -1 \times 0 + 0 \times 100 + 1 \times 100 = 300
 \end{aligned}$$

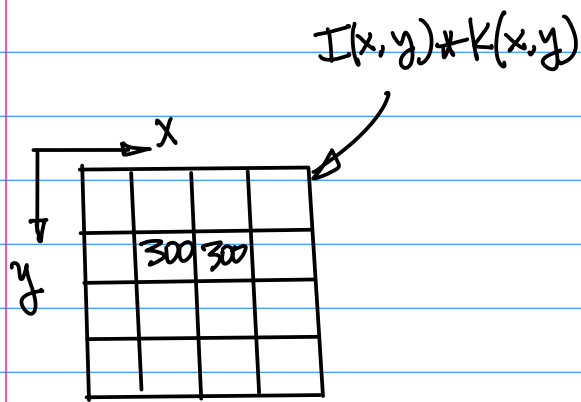


Fig. 3F

Step 3.

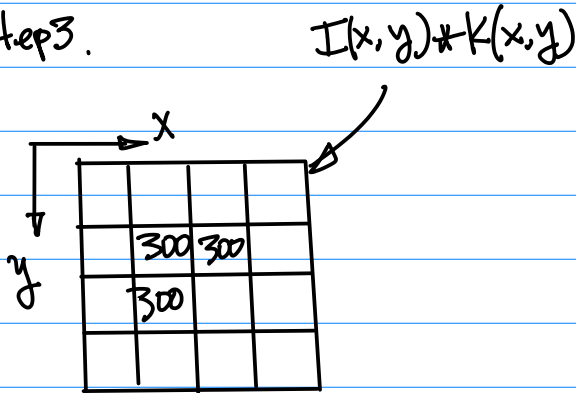


Fig. 3G

Step 3.

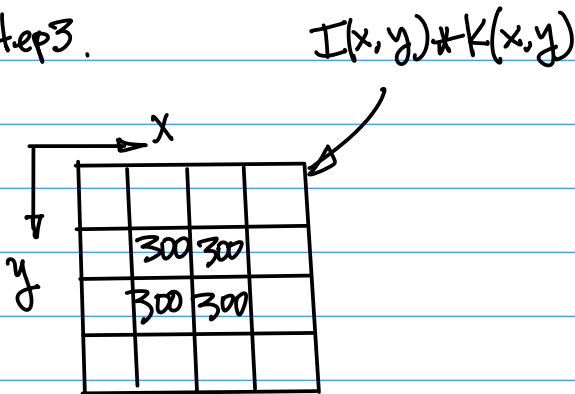


Fig. 3H

Note: 2D Convolution (Continuous Case)

given Image $f(x,y)$
a Kernel $h(x,y)$

$$\iint_{\mathbb{R}^2} f(u,v) h(x-u, y-v) du dv \quad \dots (2)$$

August 31 (Th).

Note: 1^o Honestly Pledge on CANVAS,
Signed & due this Friday.
(ON CANVAS). Homework ON CANVAS

2^o Will post Anaconda Installation
& OpenCV Installation, display
an image.

(1) Screen Capture of the Activated
CONDA Environment, with personal
identifier;

(2) Screen Capture of the OpenCV
Display with P.I.D.

Sample Code for the display to
Be Re-posted on github, 2023F

Example: Theoretical Aspects
for 2D Convolution.

Ref: 1^o PPT on the github.



2^o 2023S-100-note-...
2023-03-21.pdf

$$\iint_{\Sigma} f(u,v) h(x-u, y-v) du dv \quad (1)$$

Image

Kernel

Note 1. Σ : Image Plane(s)

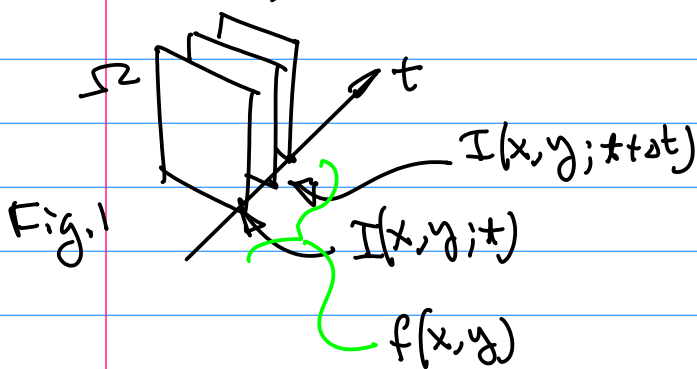
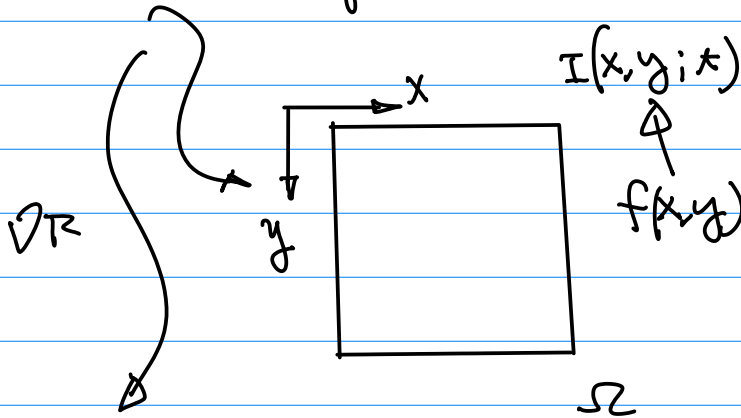


Fig. 1

Note 2: Kernel

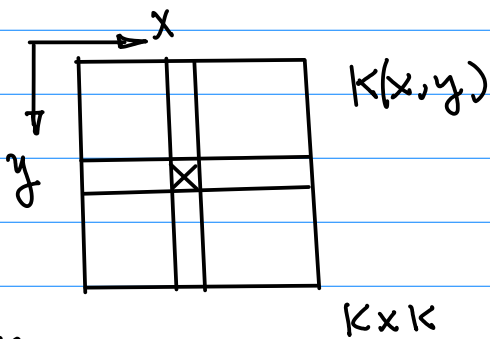
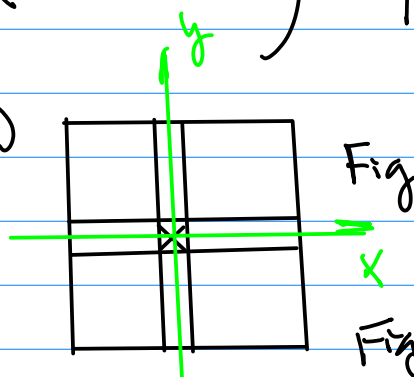


Fig. 2a

①

$K(-x, y)$

Flip the Kernel
w.r.t y -axis



Fig

Fig. 2b

② Flip the Kernel
w.r.t. x -axis.
(Since $K(-x, y)$ changed
to $K(-x, -y)$)

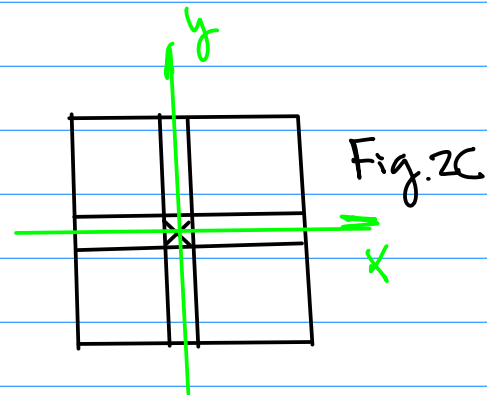
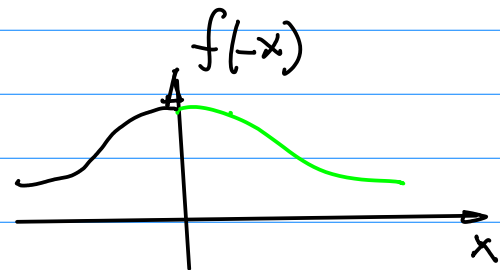
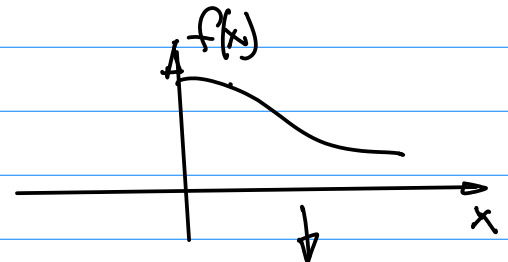
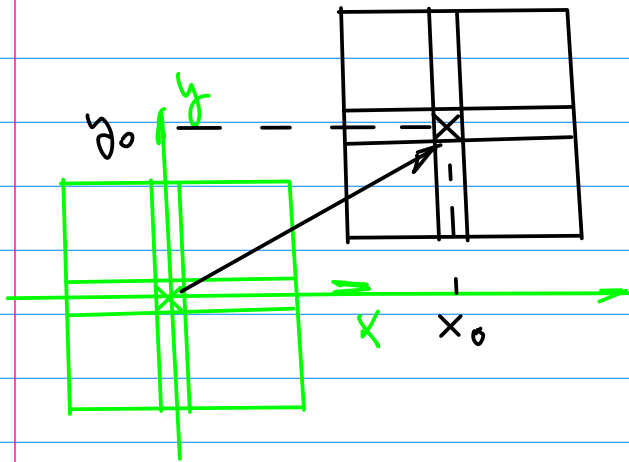


Fig. 2c

Note: for 1D Case



Next, make $K(-x, -y)$ to
 $K(x-x_0, y-y_0)$



Note: $\frac{k-1}{2}$ for the top Rows
 $\frac{k-1}{2}$ for the Bottom Rows
 Lost
 $2 \times \frac{k-1}{2} = k-1$

Similarly for the col.s.

Therefore,
 Input Dimension $M \times N$ \rightarrow Output After Convolution
 $m - (k-1) \times$
 $N - (k-1)$
 $\dots (3)$

Fig. 2d

Now, for Discrete Image (OR Digital Feature plane)

Replace $\iint \rightarrow$ by $\sum_{k_1} \sum_{k_2}$

$$c(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} a(k_1, k_2) b(n_1 - k_1, n_2 - k_2)$$

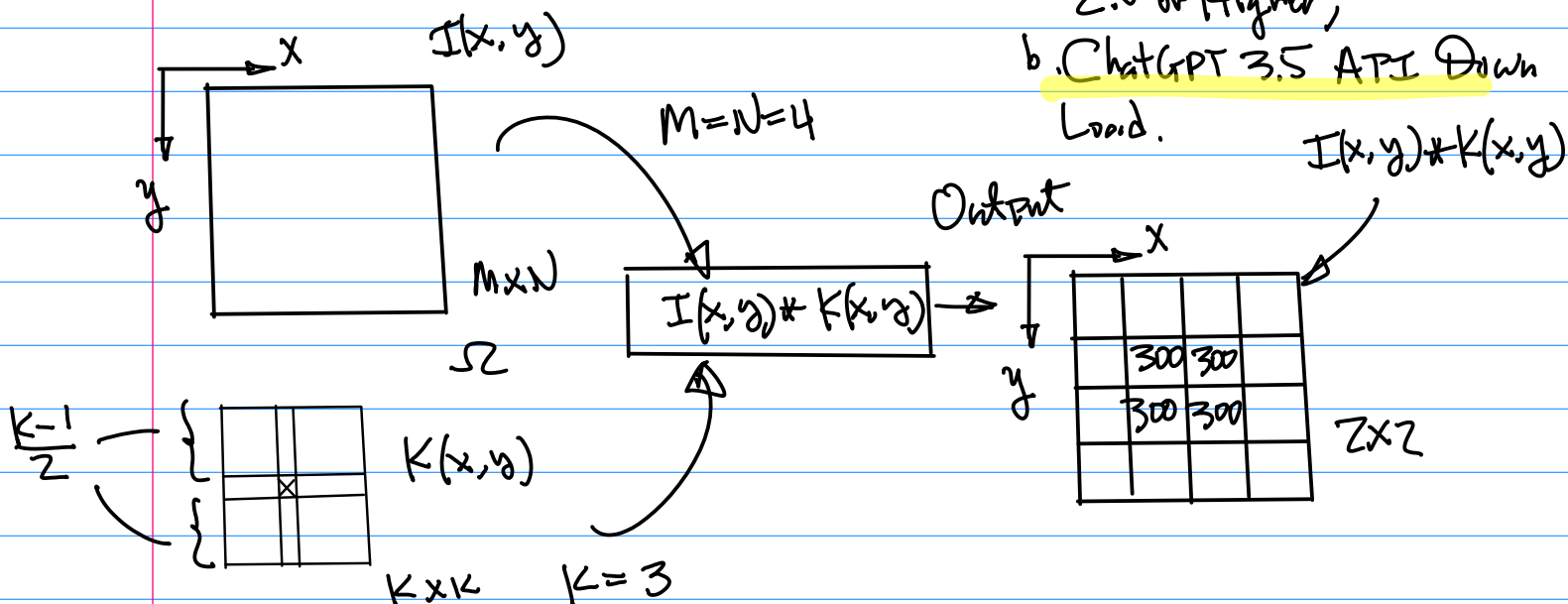
Image Kernel ... (2)

Sept. 5 (Tue)

Note: 1st CANVAS OPENCV, Anaconda Installation;
 2nd Homework Coming On Thursday.

- a. Installation of T.F. Version 2.0 or Higher;
- b. ChatGPT 3.5 API Down Load.

Dimension Change Due to the Convolution.



Example: Continuation on
Convolutional Layers and
Architecture for Handwritten
Digits Recognition.
Based on Eq (3), pp.9.

$$M \times N \text{ Image} \xrightarrow{K \times K \text{ Convolution}} M - (K - 1) \times N - (K - 1)$$

From MNIST Architecture,
the Input Image $I(x, y)$ with
 $M \times N$ ($M = N = 28$) the
Output feature Layers Resolution
 24×24 . $\Rightarrow K = (28 - 24) + 1$
Resolution difference.

Now, the total No. of feature
layers = n

Since one Convolution (e.g.
Using one $K \times K$ Kernel
to perform Convolution)
produces 1 feature Layer,

Therefore
 n Convolutional feature
Layers is the
result of n Convolutions,
e.g. n different
Kernels.

Note: Multiple feature Layers
are due to the inspiration
of Human Visual Perception
System, e.g. "Early" Vision
System.

Log Laplace of Gaussian.
Gabor Operators \nearrow

$$G(x, y) = \frac{1}{\sqrt{\pi \sigma_x \sigma_y}} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{(y-\mu_y)^2}{2\sigma_y^2}} \dots (1)$$

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \dots (2)$$

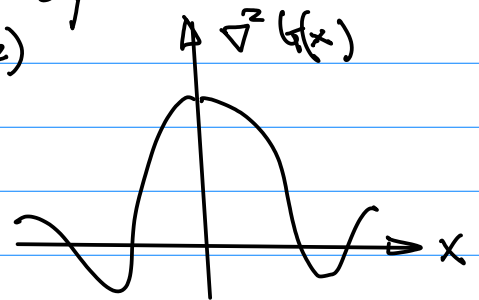


Fig. 1.

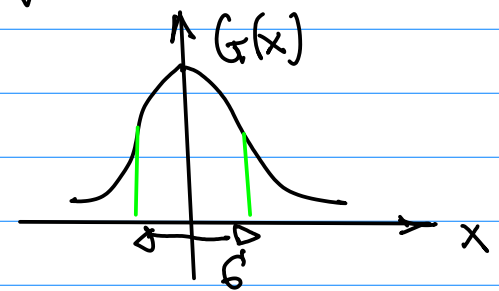


Fig. 2

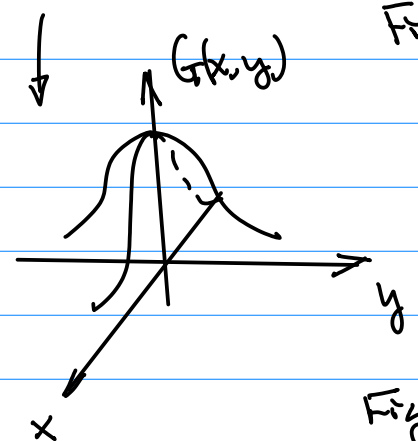


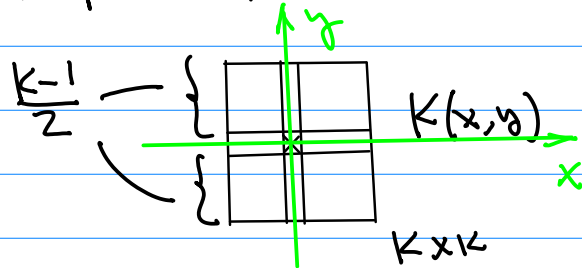
Fig. 3

CMPE258

F2023

11/

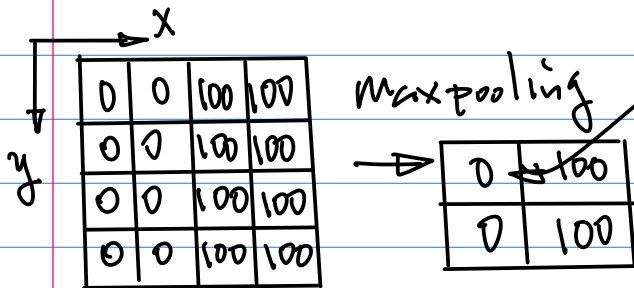
Take $G(x, y)$ in Fig. 3,
map it to $K \times K$.



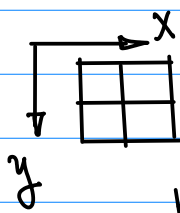
n | Kernels Produce n | Layers.

Now, consider a technique
for feature Reduction,

Given an image $I(x, y)$



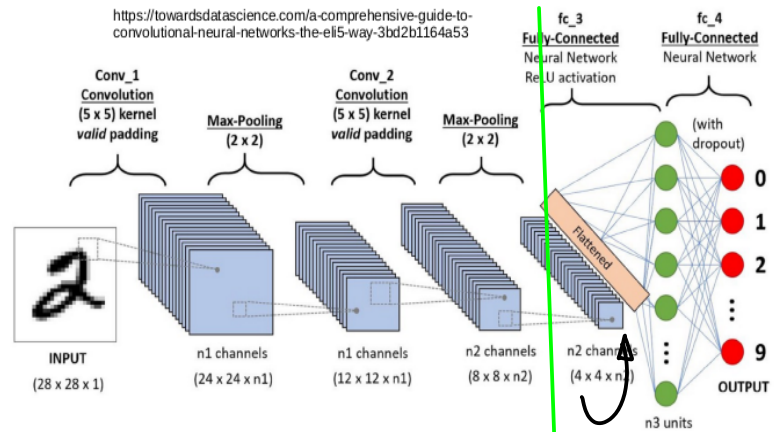
Feature Layer



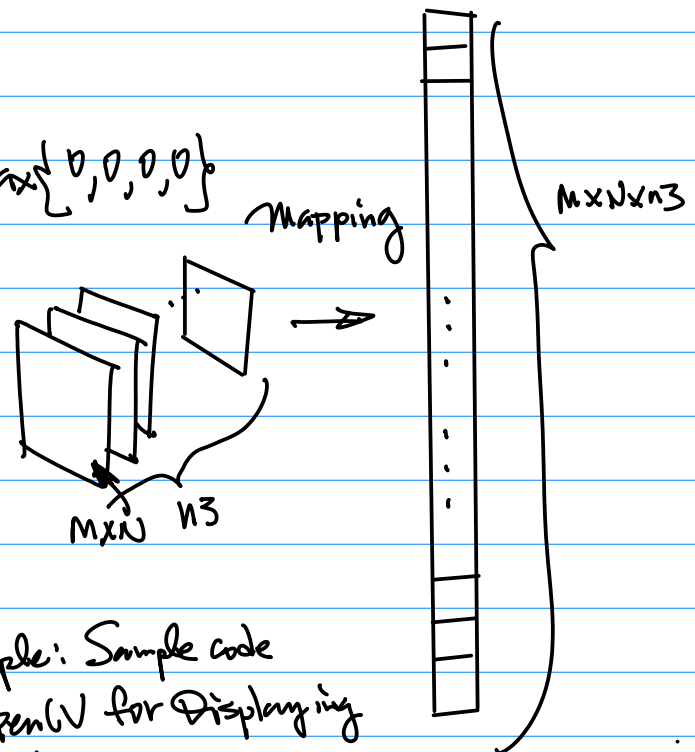
where $K=2$

Now, consider the Architecture of MNIST

Illustration of A CNN for Digits Recognition



Consider Re-arrange 2D feature layers
into 1D
Neurons.

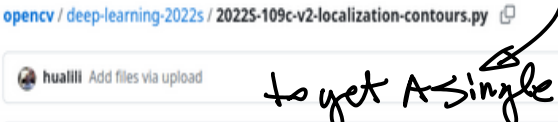


Example: Sample code
OpenCV for Displaying
an image

cv2.imread(imageName, cv2.IMREAD_COLOR)
cv2.imshow(window name, src)

Sept. 7 (Th).

Note: 1st About OpenCV
Reference/Sample
Code.

- a) On github. for Video Capture

 to get A Single Frame.

- b) Sample for Single frame, e.g.
 .jpg, .png etc Image Input.

2022S-104d ~ Python Code

- c) Script for Conda Environment
 On github:

2022S-104b ~
 " -104c ~

- d) Readme for Creating Conda
 Environment.

Note 2. Homework Due A week
 from today (Sept. 17, Sun)

Requirements:

- a) Installation of T.F. Version 2.0
 or higher
 b) Screen Capture that shows
 T.F. installed Successfully.
 (with Personal Identifier)

Submission ON CANVAS.

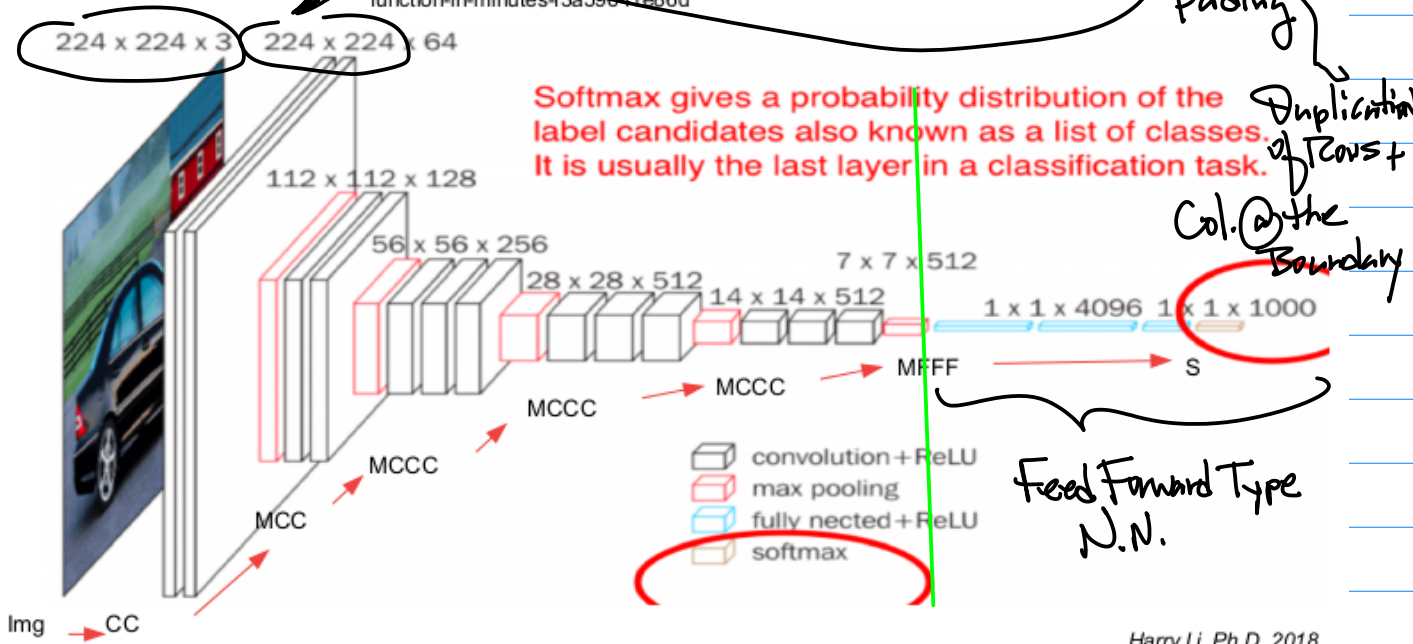
Example: Convolutional Layer, Architecture Analysis.

Note 1. Architecture; Note 2:

Padding v.s. No

Architecture Example VGG16

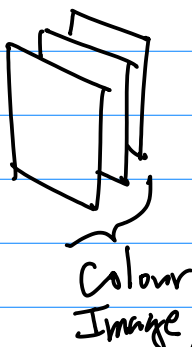
<https://medium.com/data-science-bootcamp/understand-the-softmax-function-in-minutes-f3a59641e86d>



Harry Li, Ph.D. 2018

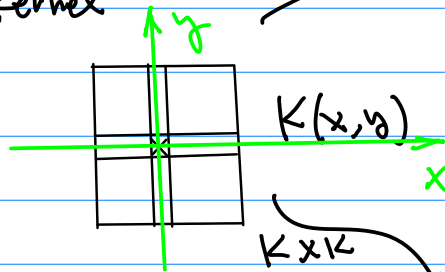
Note 3: Convolution Discussion.

One Example



$$I(x, y) = (r(x, y), g(x, y), b(x, y))$$

Kernel



	x		
y	-1	0	+1
	-1	0	+1
	-1	0	+1

$$\text{Log } \nabla^2 G(x, y) \rightarrow \text{P.P.D.}$$

$$\nabla^2 G(x, y; \mu_x, \mu_y, \sigma_x, \sigma_y)$$

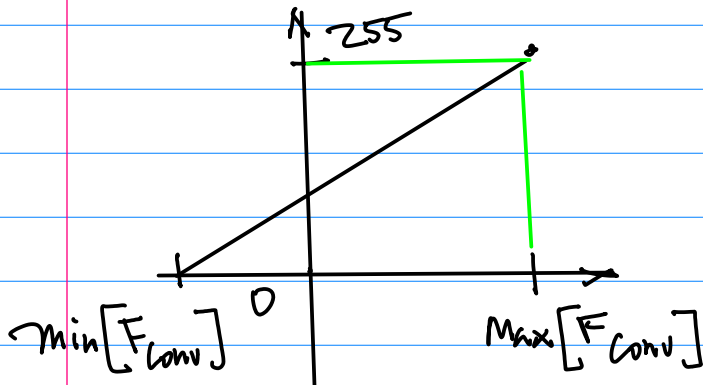
$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) G(x, y; \mu_x, \mu_y, \sigma_x, \sigma_y)$$

To perform Convolution. $\rightarrow K(x, y)$

First, kernel is one channel in this case;

Then, Convolutions. $r(x, y) * K(x, y), g(x, y) * K(x, y), b(x, y) * K(x, y)$

$$F_{\text{conv}}(x, y) = \frac{1}{3} [r(x, y) * k(x, y) + g(x, y) * k(x, y) + b(x, y) * k(x, y)]$$



Consider A Design of S.I.D
Recognition System Using T.F &
MNIST CNN as a Processing
Engine.

Requirements:

1° Live CAM Input,

720P or 1080P

1280x720 (MxN)

Col.

Row.

1920x1080

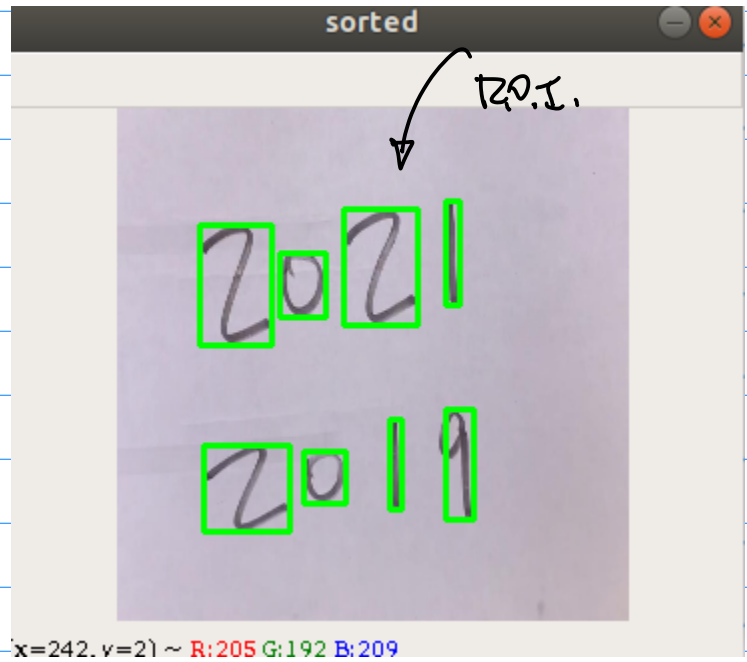
(MxN)

Col.

Row.

2° Printer Paper (Blank/White)
Black mark to Write
4 Digits

3° Localize R.O.I. On Each
Digit



Region of Interests.