



with tensorflow

구현을 위한 딥러닝

■ 고려대학교 물리학과 한승희

■ mod96@naver.com

Contents

1

Traditional CV

2

Convolutional Layer

3

Why CNN?

4

Bias - Variance Tradeoff

5

Useful Things

6

Implementation

Contents

1

Traditional CV

2

Convolutional Layer

3

Why CNN?

4

Bias - Variance Tradeoff

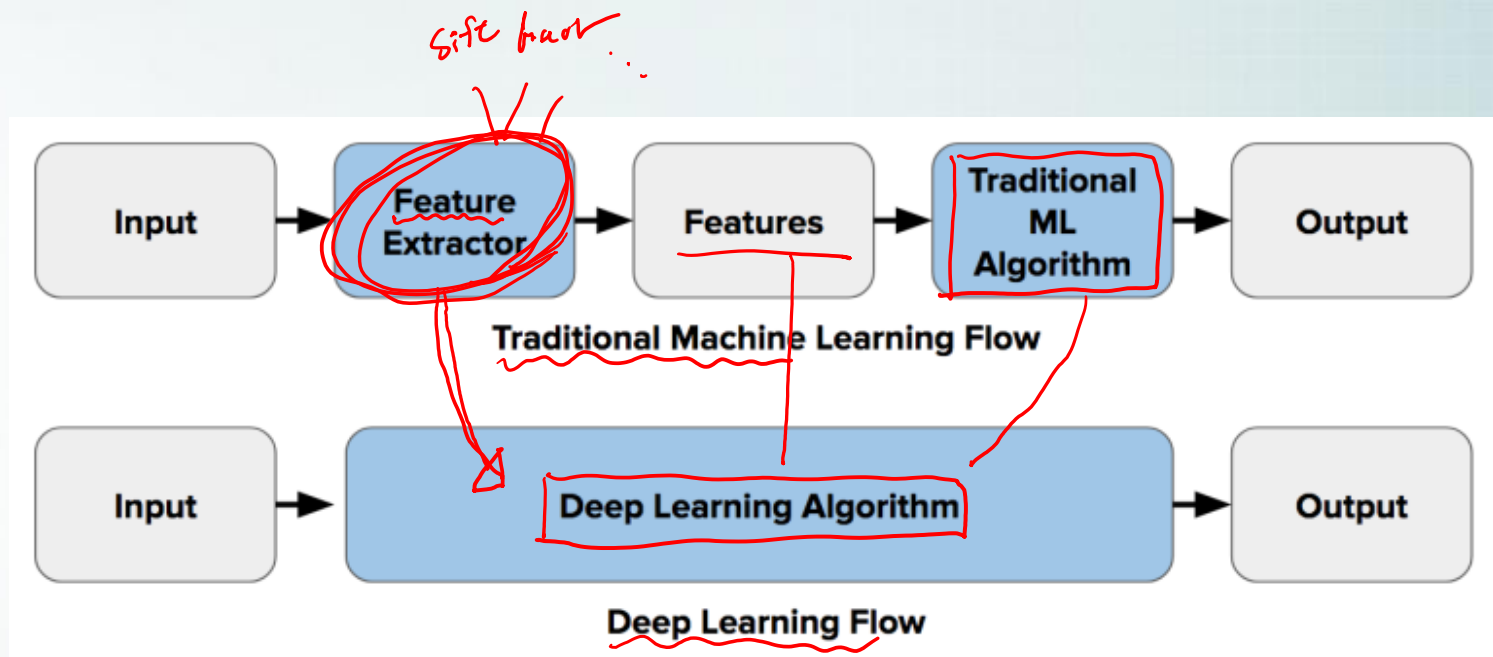
5

Useful Things

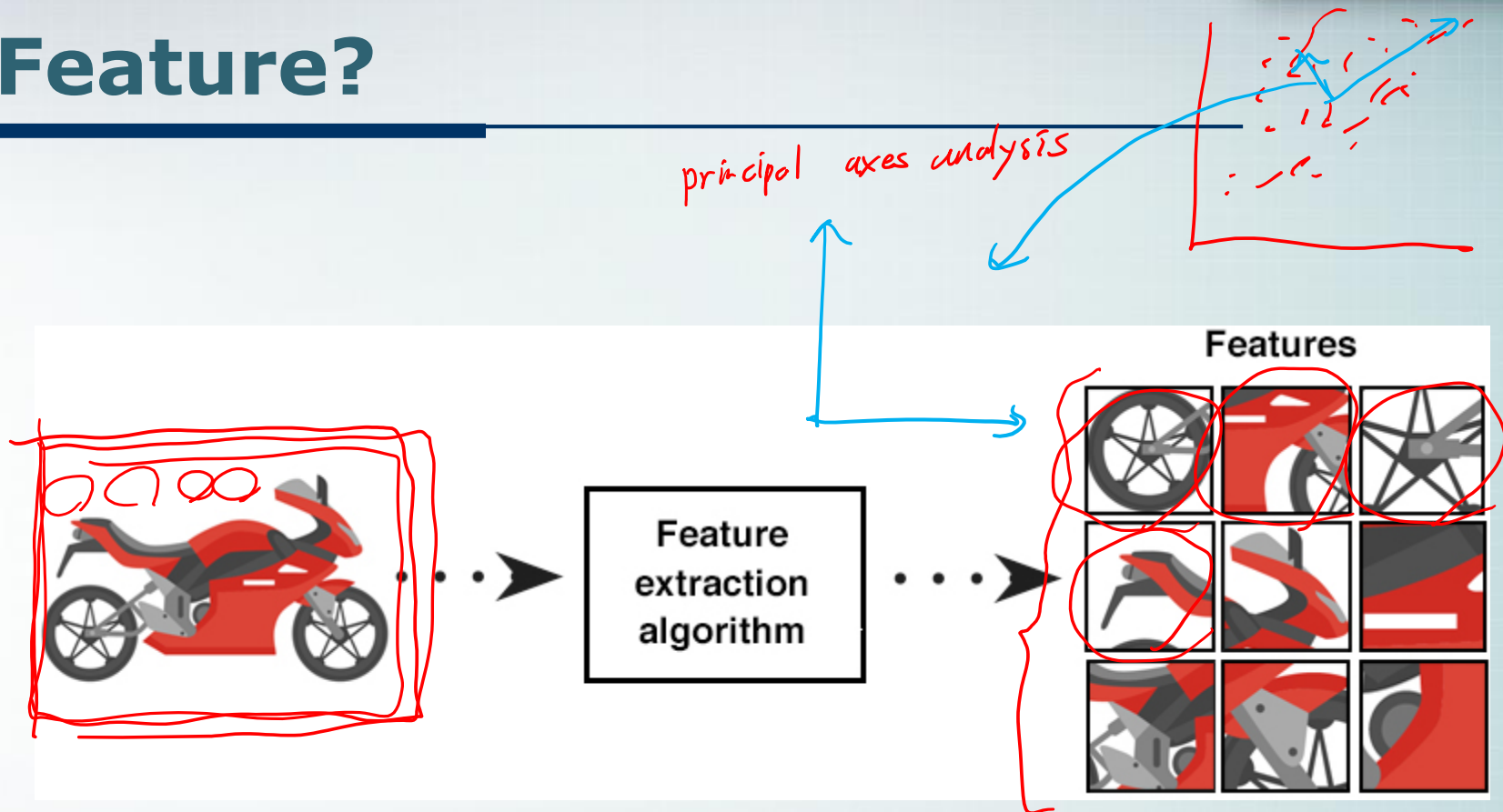
6

Implementation

Traditional vs DL in CV



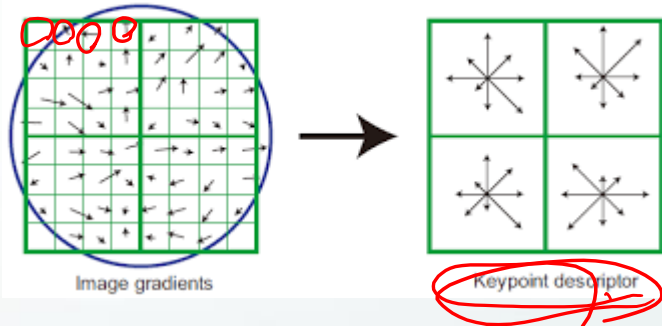
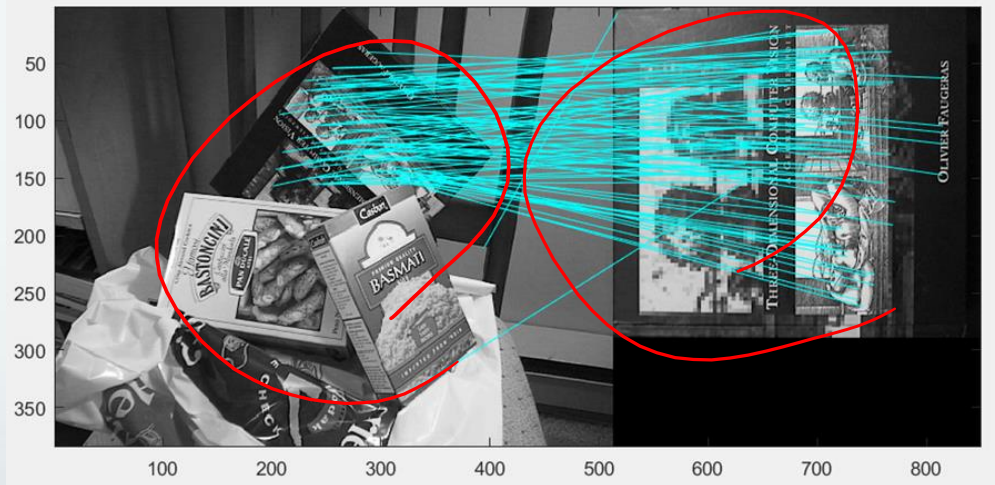
Feature?



- ❖ EX> Color, Global Shape(PCA space), Local shape(shape context), Texture(Filter banks)

Feature?

❖ SIFT, SURF, BRIEF,
ORB, ...



Feature?

❖ HOG, ...



gradient
orientation



gradient
magnitude



histogram of
gradient orientation

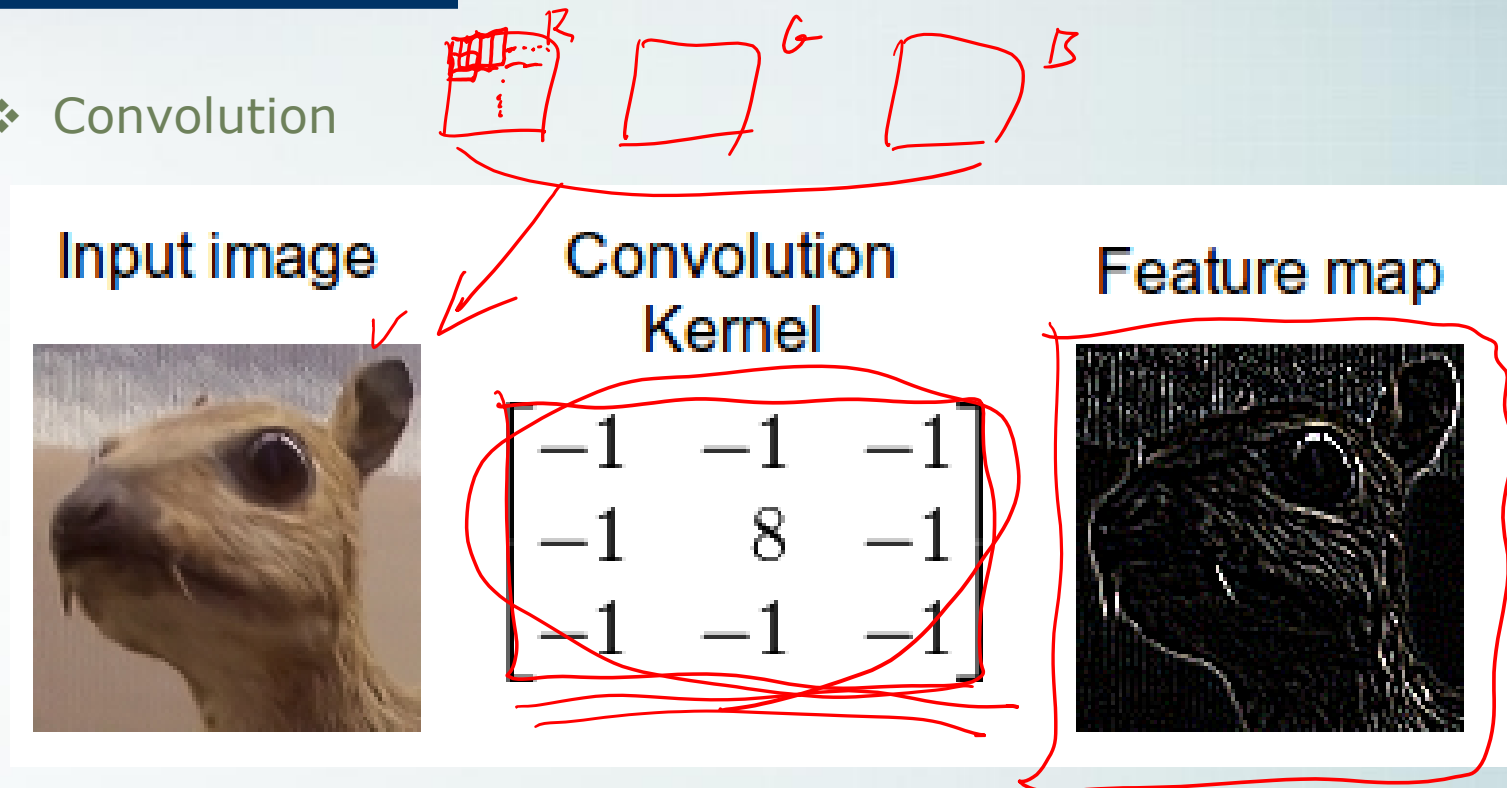
Feature?

❖ MCT, Haar feature, LBP, ..



Feature?

❖ Convolution

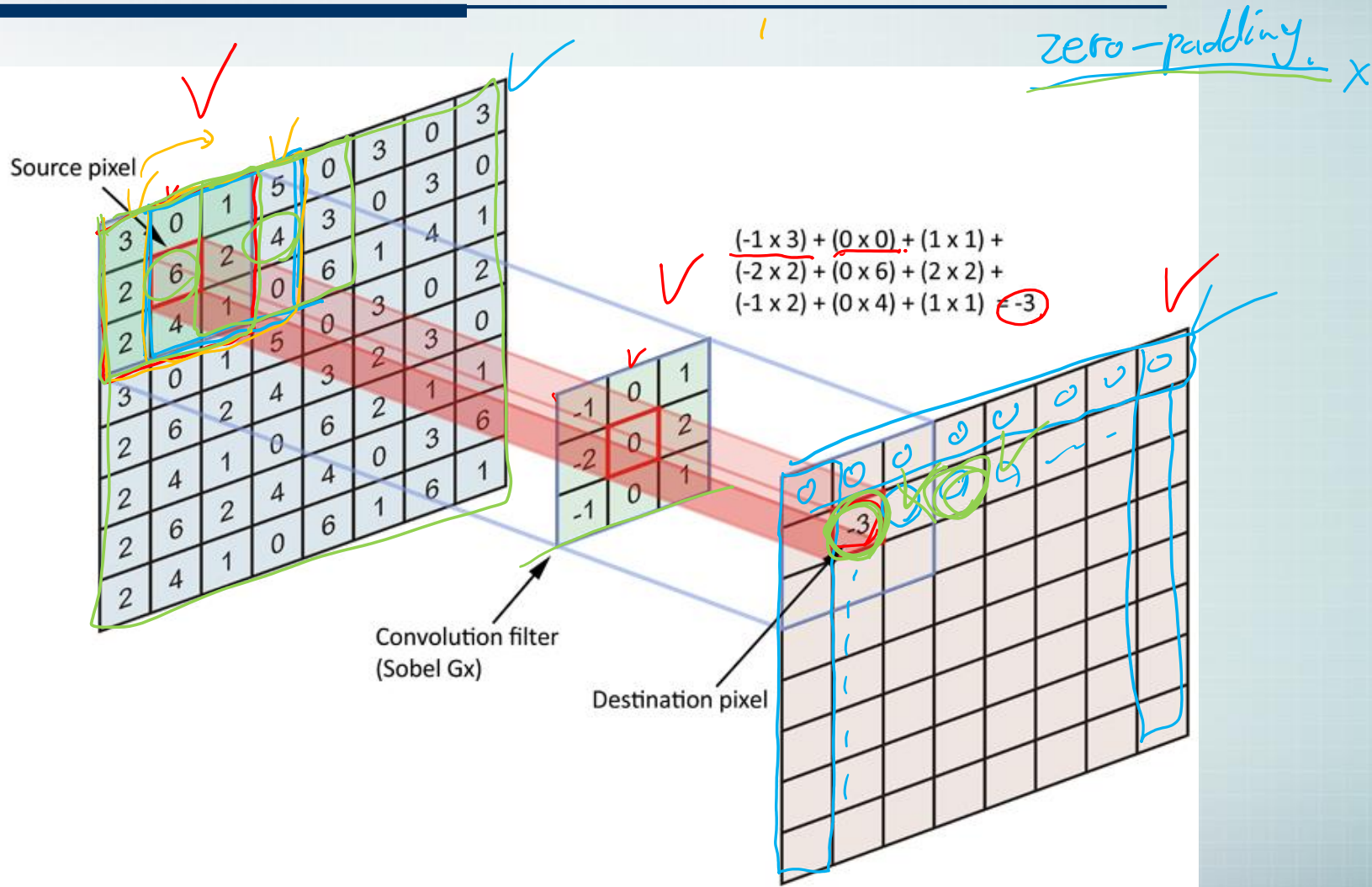


- [https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))

Contents

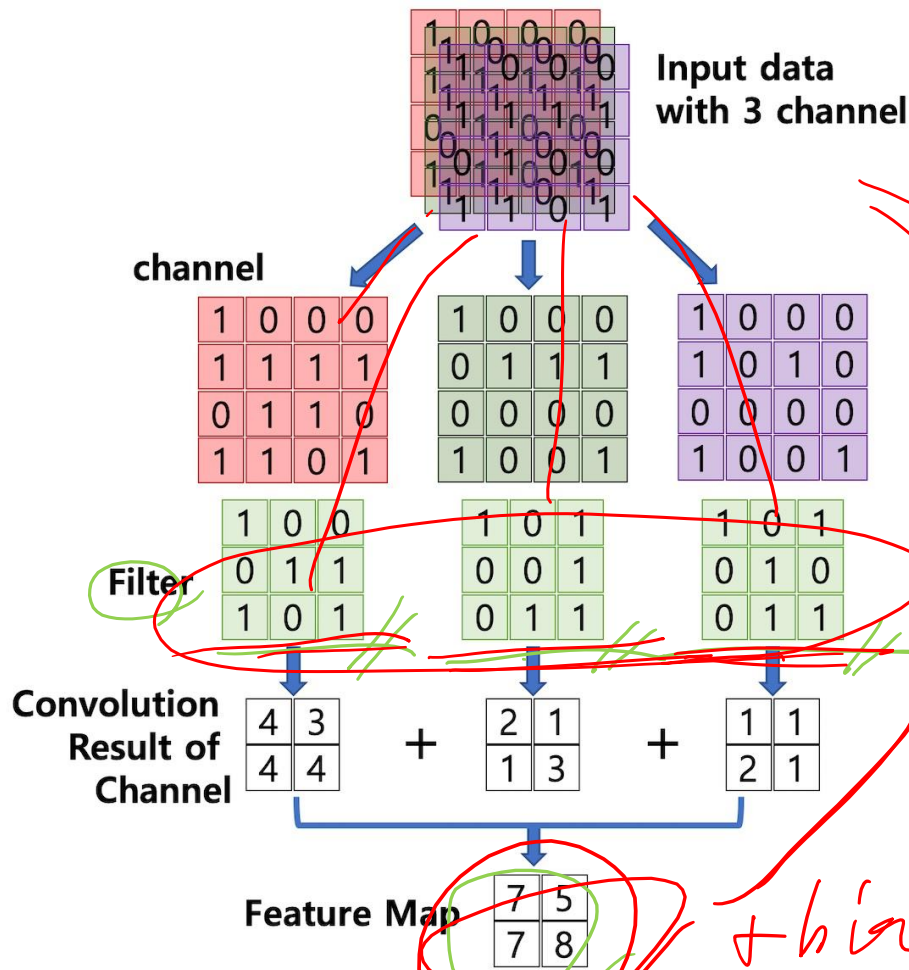
- 1 Traditional CV
- 2 Convolutional Layer
- 3 Why CNN?
- 4 Bias - Variance Tradeoff
- 5 Useful Things
- 6 Implementation

Kernel (filter), Stride, Padding



Kernel (filter)

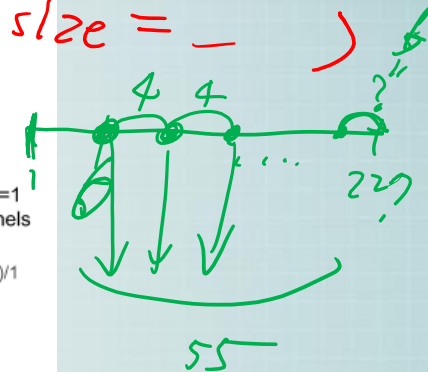
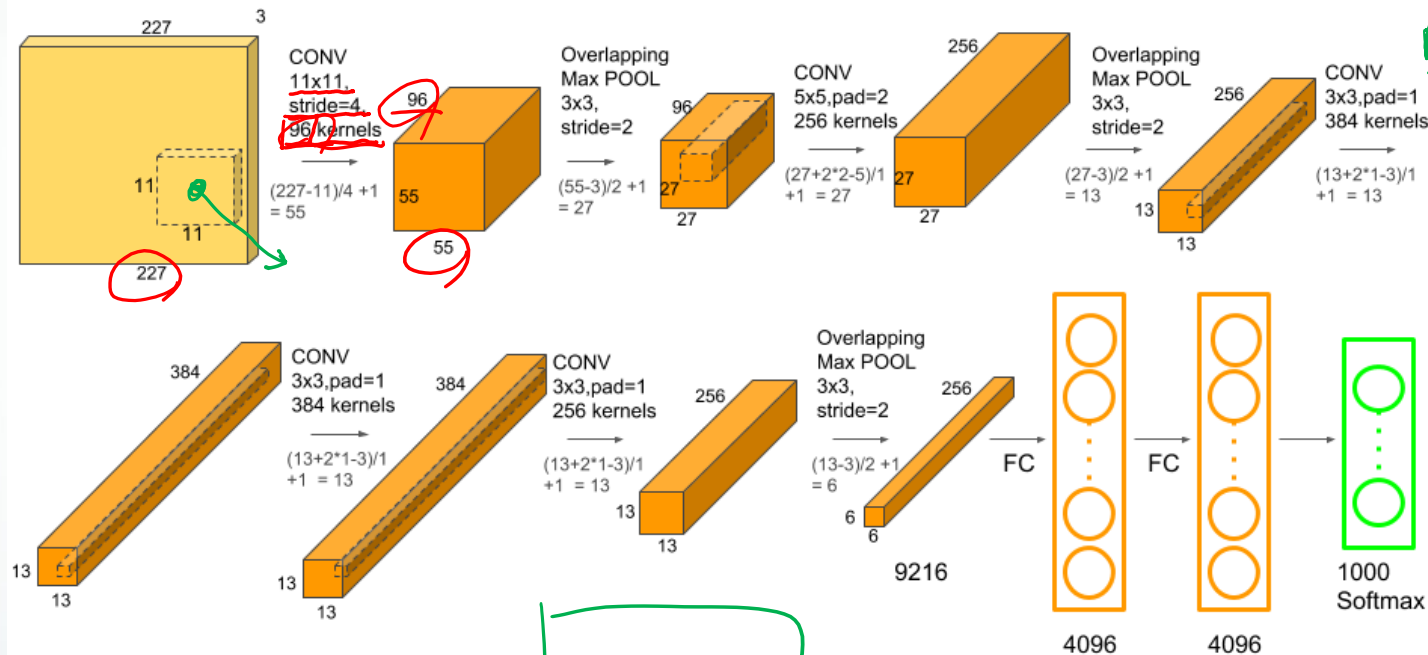
R, G, B



Example

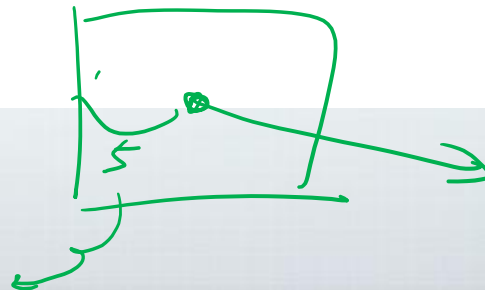
$227 \times 227 \times 3$

Conv2D (\square , stride = , padding = , kernel_size =)



$$4(N-1) + \dots + ? = 227$$

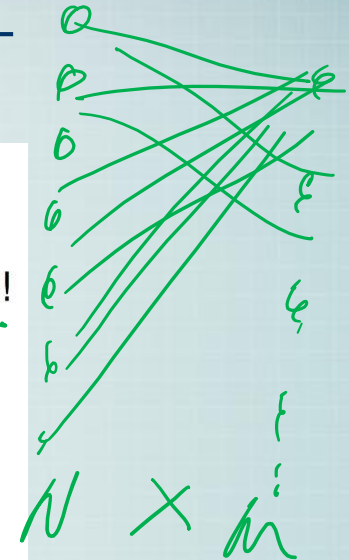
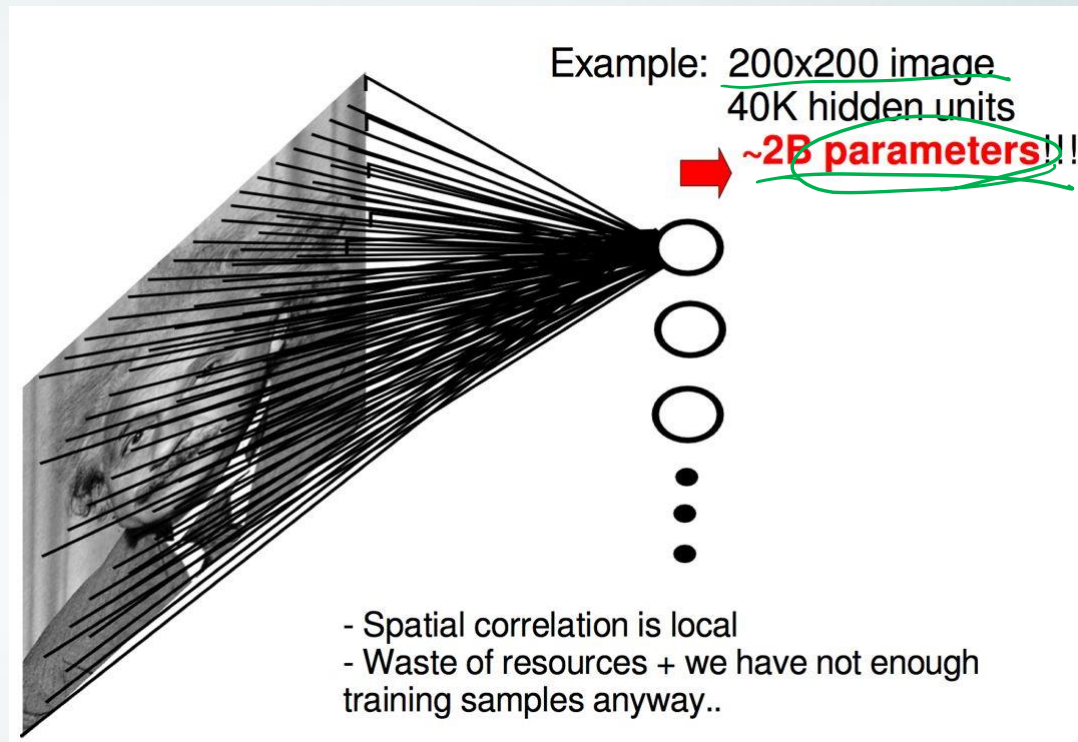
$$4(N-1) = 227 - ?$$



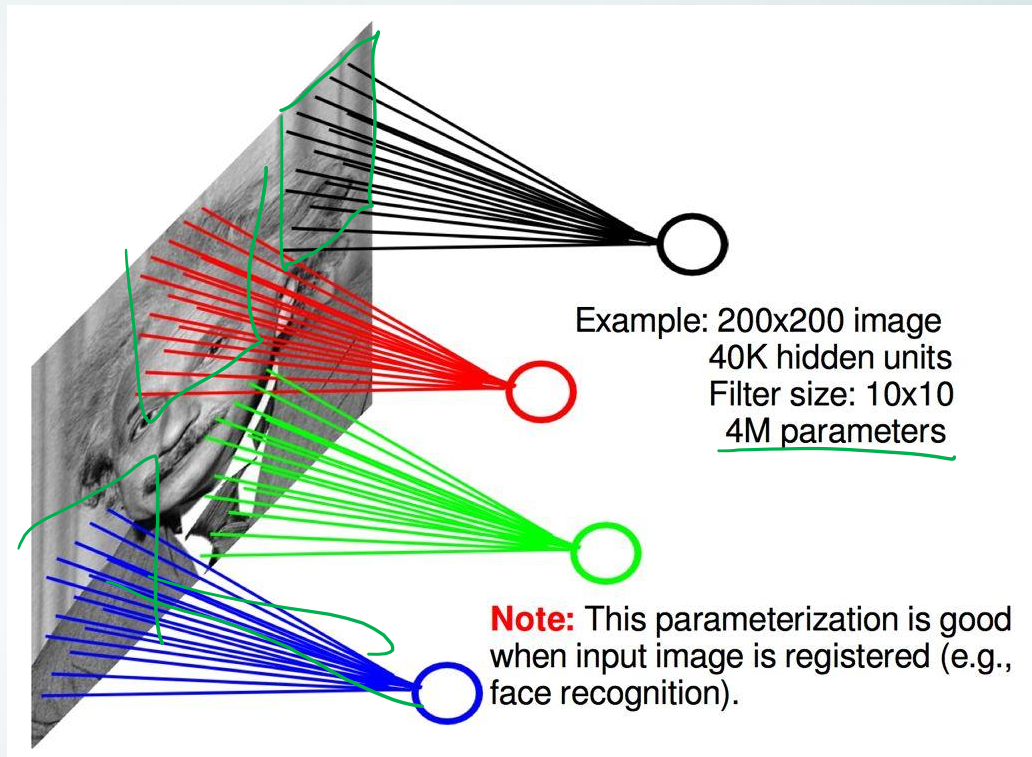
Contents

- 1 Traditional CV
- 2 Convolutional Layer
- 3 **Why CNN?**
- 4 Bias - Variance Tradeoff
- 5 Useful Things
- 6 Implementation

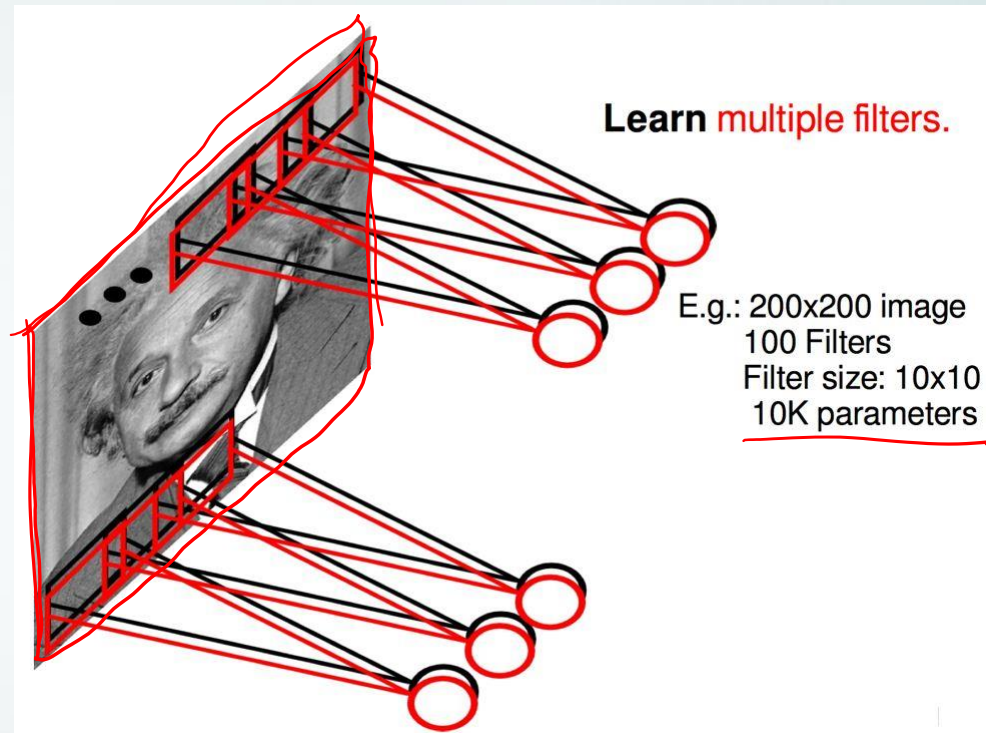
If Fully Connected (Dense)...



If Locally Connected



CNN



Contents

- 1 Traditional CV
- 2 Convolutional Layer
- 3 Why CNN?
- 4 **Bias - Variance Tradeoff**
- 5 Useful Things
- 6 Implementation

As before..

- ❖ Batch Normalization
- ❖ Dropout
- ❖ Weight Initialization methods
- ❖ Regularization
- ❖ ...

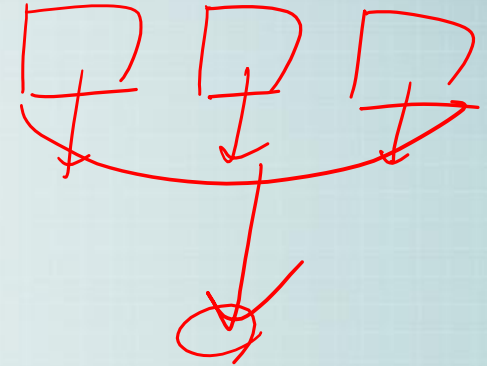
Pooling

12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

2×2 Max-Pool

20	30
112	37

- ❖ MaxPool, AveragePooling
- ❖ GlobalMaxPool, GlobalAveragePooling



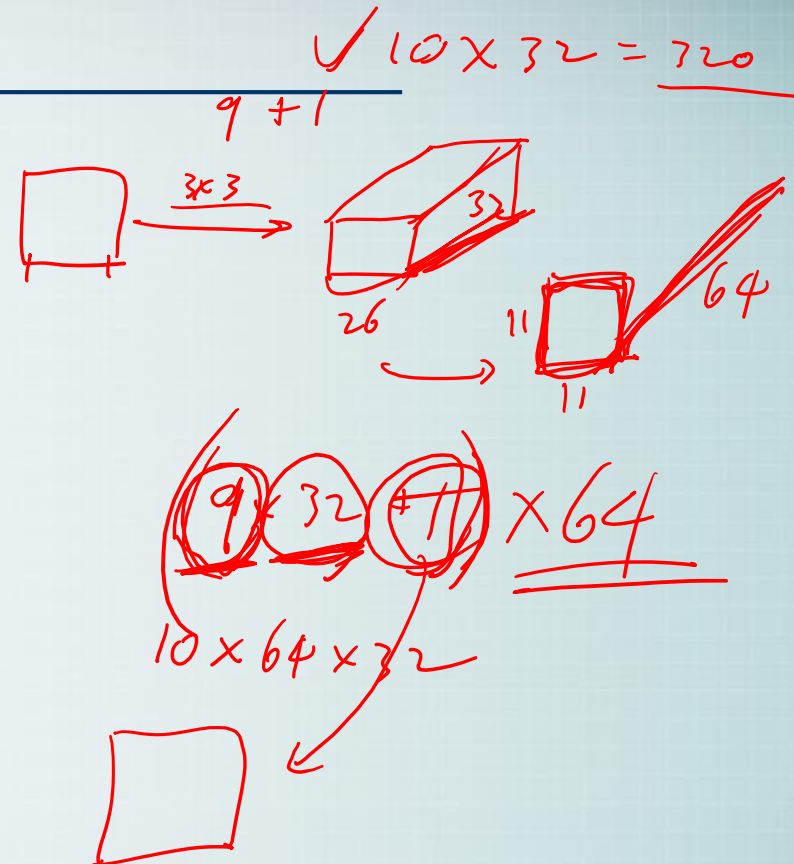
Sample Calculation

```

model = keras.Sequential(
    [
        keras.Input(shape=input_shape),
        layers.Conv2D(32, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Conv2D(64, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Flatten(),
        layers.Dropout(0.5),
        layers.Dense(num_classes, activation="softmax"),
    ]
)

```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	320
<u>max_pooling2d (MaxPooling2D)</u>	(None, 13, 13, 32)	0
conv2d_1 (Conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 64)	0
flatten (Flatten)	(None, 1600)	0
dropout (Dropout)	(None, 1600)	0
dense (Dense)	(None, 10)	16010



Contents

1

Traditional CV

2

Convolutional Layer

3

Why CNN?

4

Bias - Variance Tradeoff

5

Useful Things

6

Implementation

tf.keras.applications

- ❖ https://www.tensorflow.org/api_docs/python/tf/keras/applications

OpenCV

cv2.dnn



ImageNet

<https://paperswithcode.com/sota/image-classification-on-imagenet>

and more...

tf.keras.preprocessing.image

https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator

Contents

1

Traditional CV

2

Convolutional Layer

3

Why CNN?

4

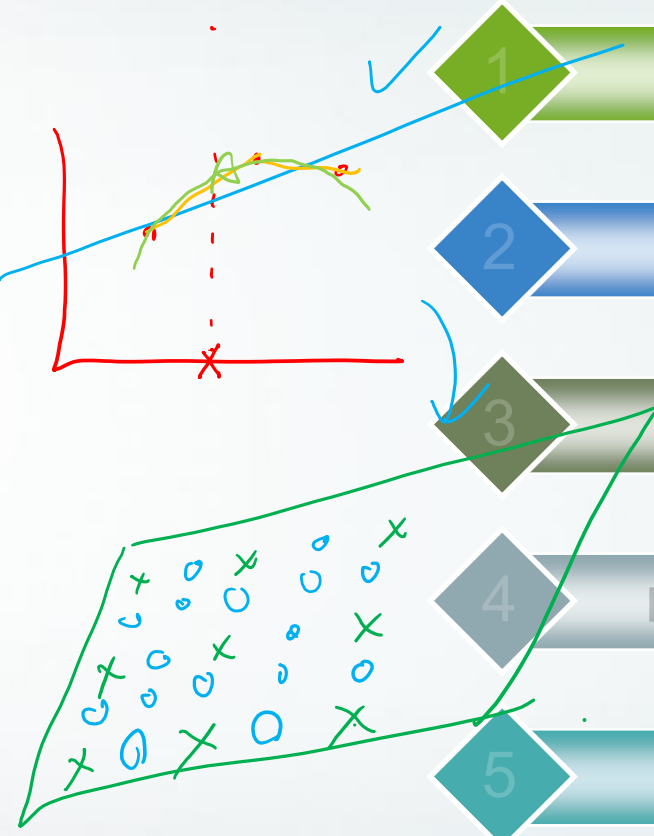
Bias - Variance Tradeoff

5

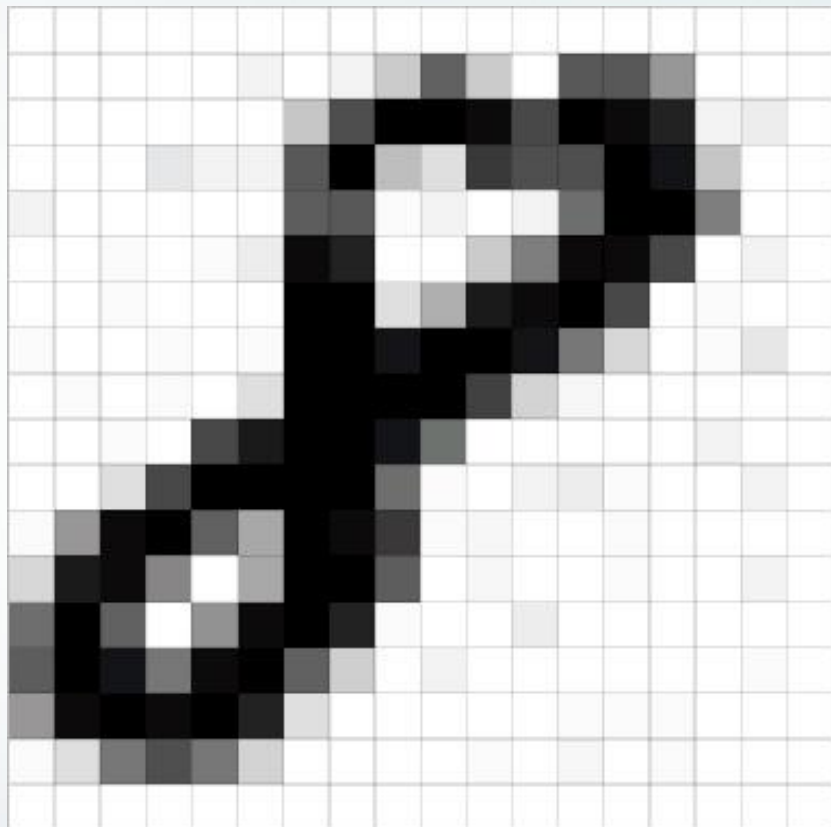
Useful Things

6

Implementation



MNIST

[illegible]

CIFAR-10

비행기



자동차



새



고양이



사슴



개



개구리



말



배



트럭

