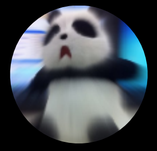
Sep 5th 2018 try! Swift NYC

#tryswiftnyc

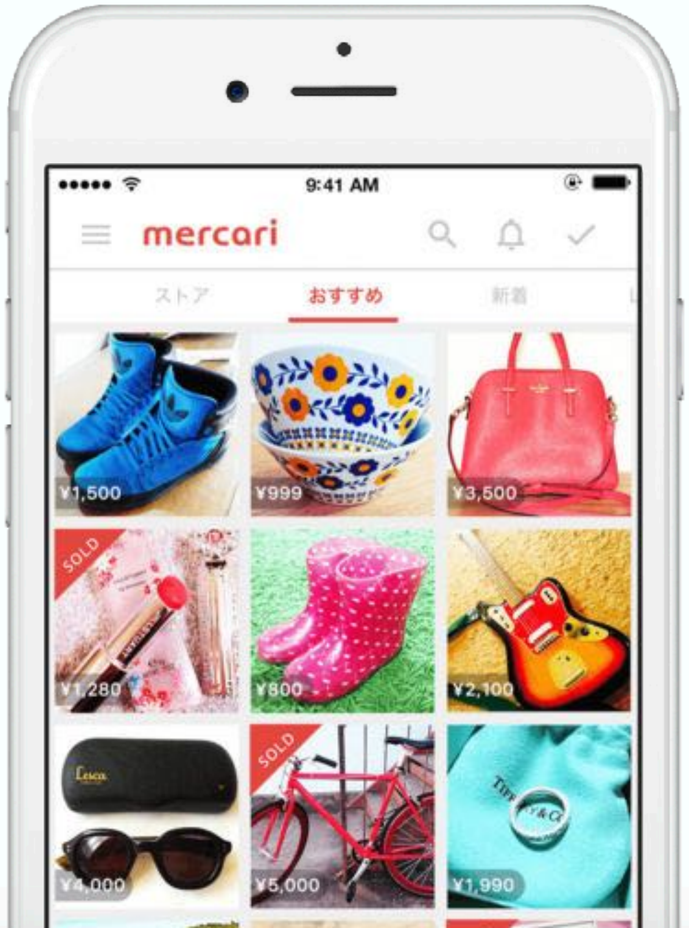
Super Resolution with CoreML



Kentaro Matsumae
Mercari

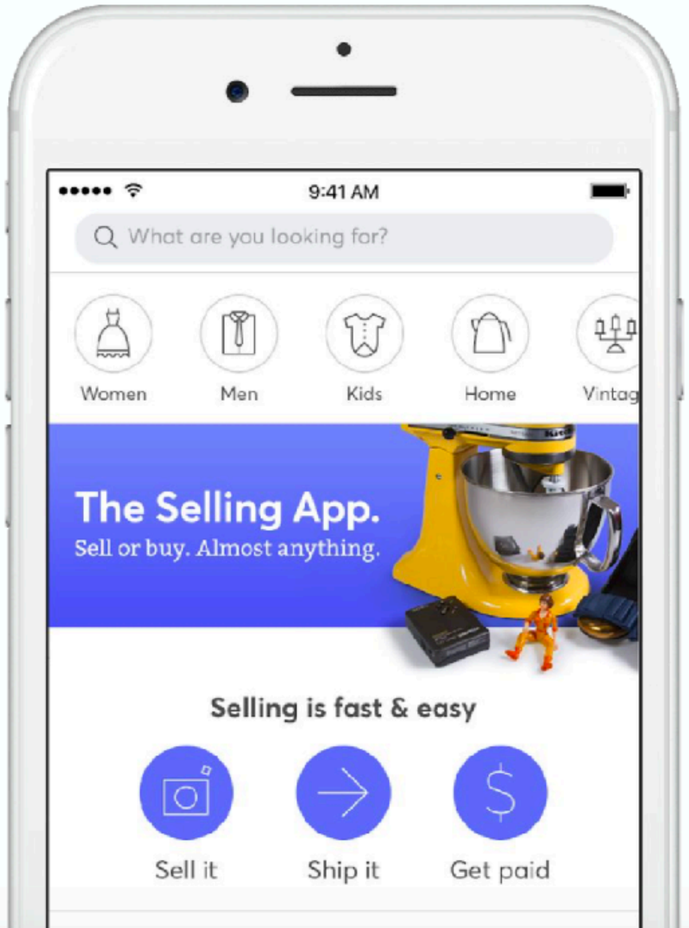
mercari

JP



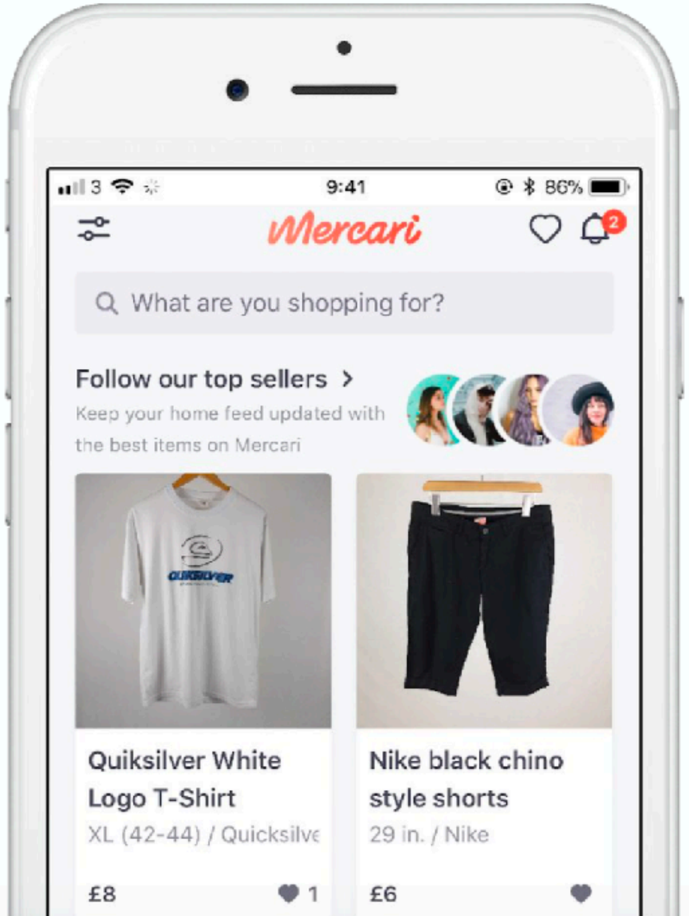
MERCARI

US



Mercari

UK



Manga App

Previous　work

ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

Manga App

Previous　work

Manga App

Previous　work

ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

Resolution vs. File Size

ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

Resolution vs. File Size

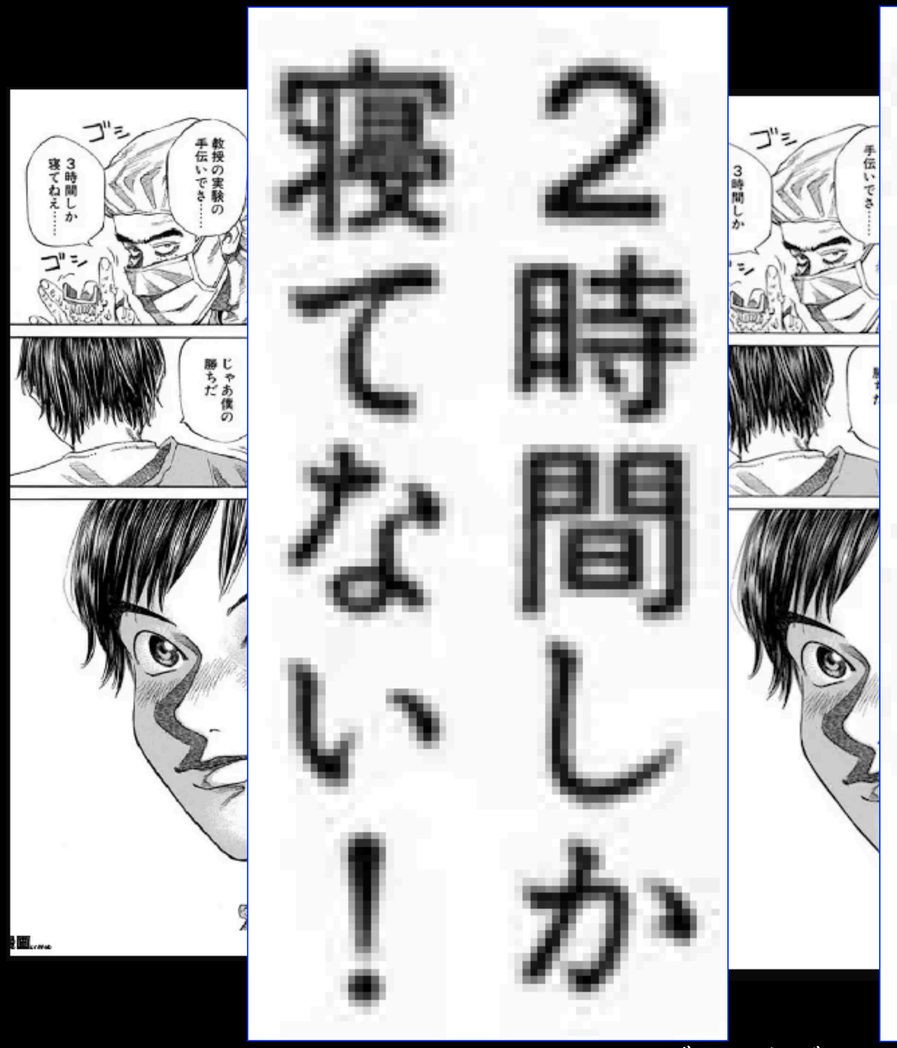
1200x800

600x400

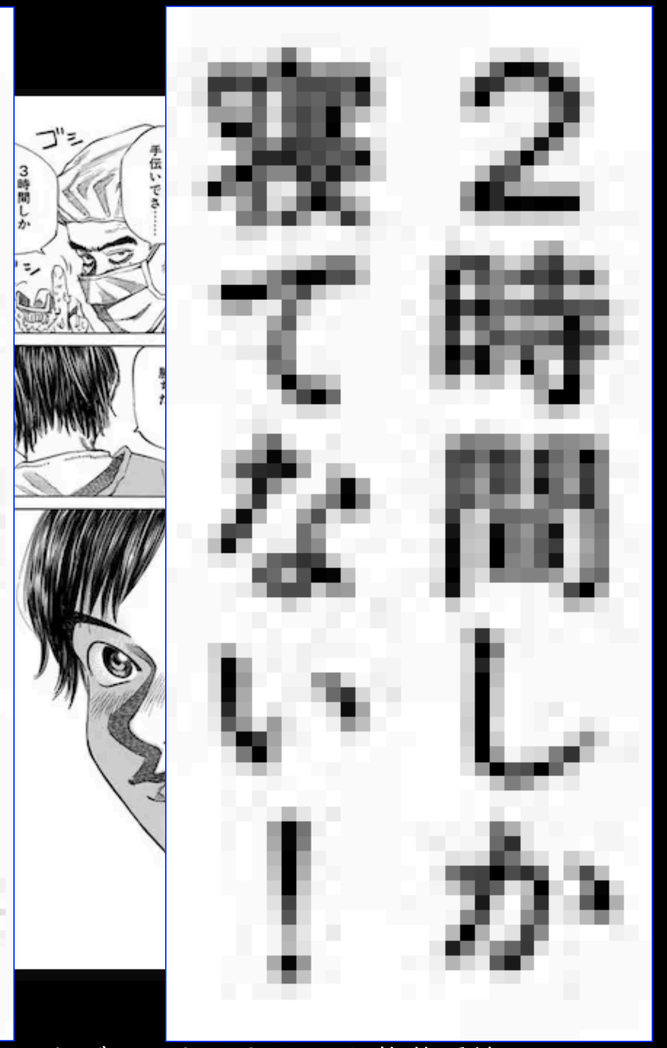
ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

Resolution vs. File Size

1200x800



600x400

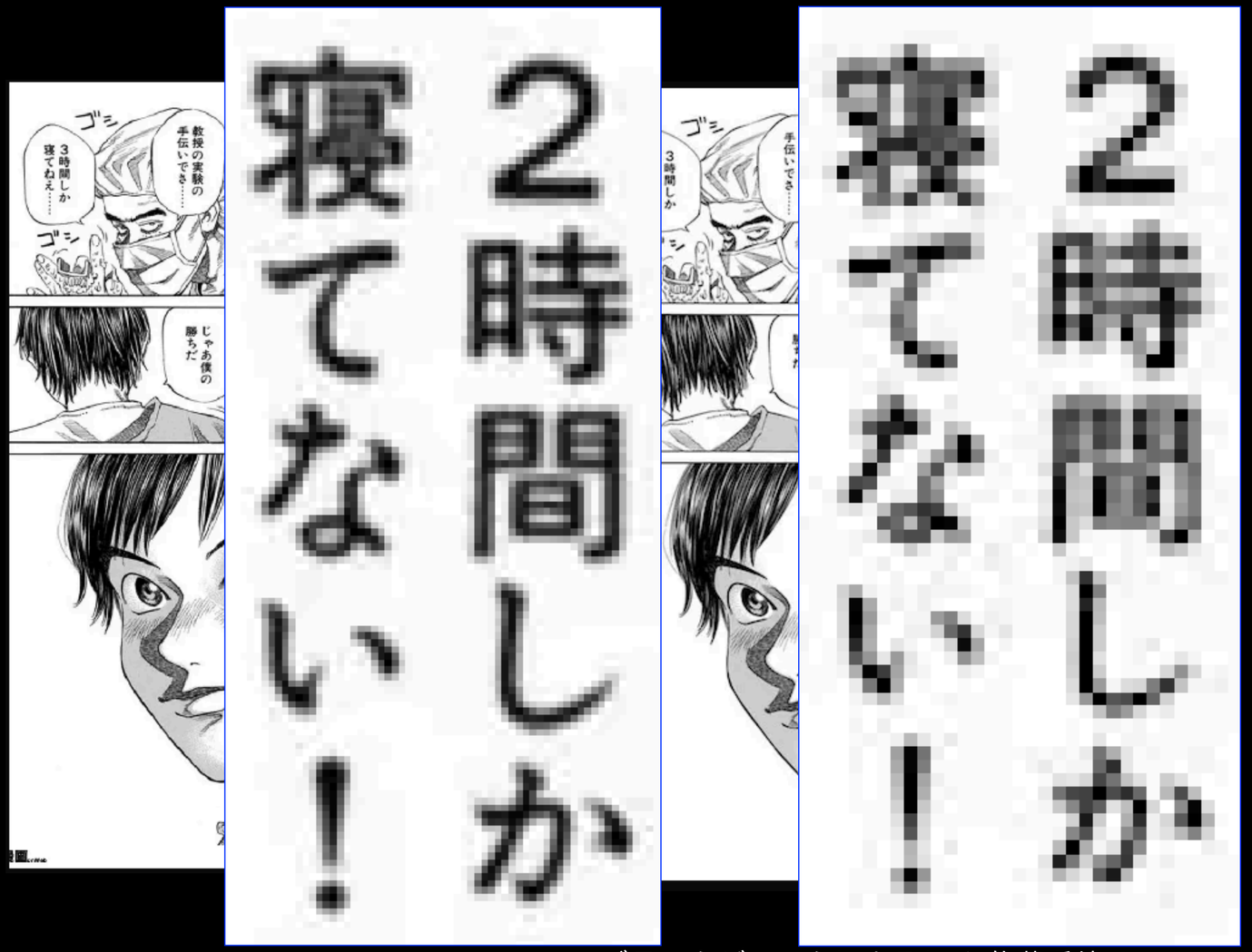


ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

Resolution vs. File Size

1200x800

100 KB



600x400

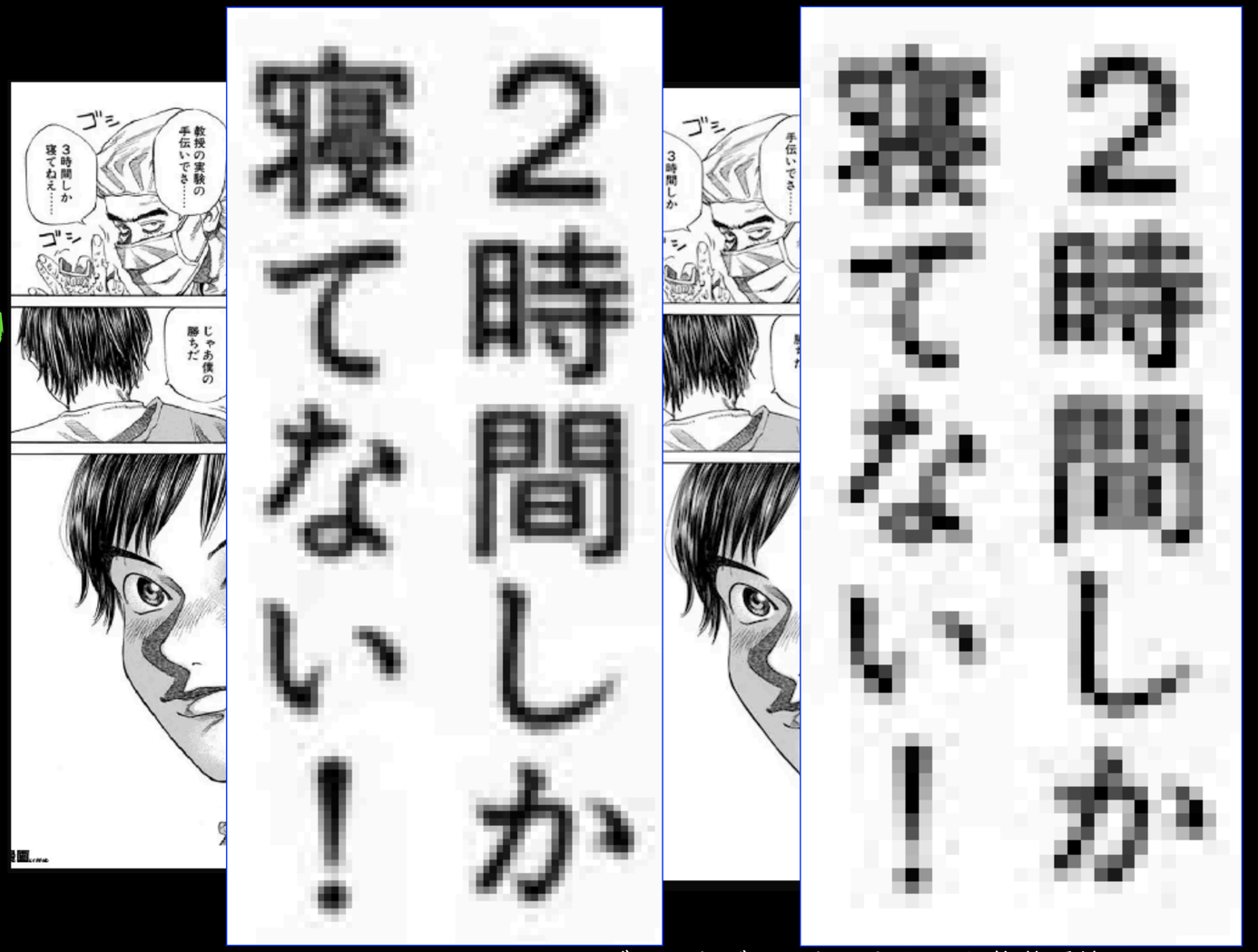
40 KB

ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

Resolution vs. File Size

1200x800

100 KB

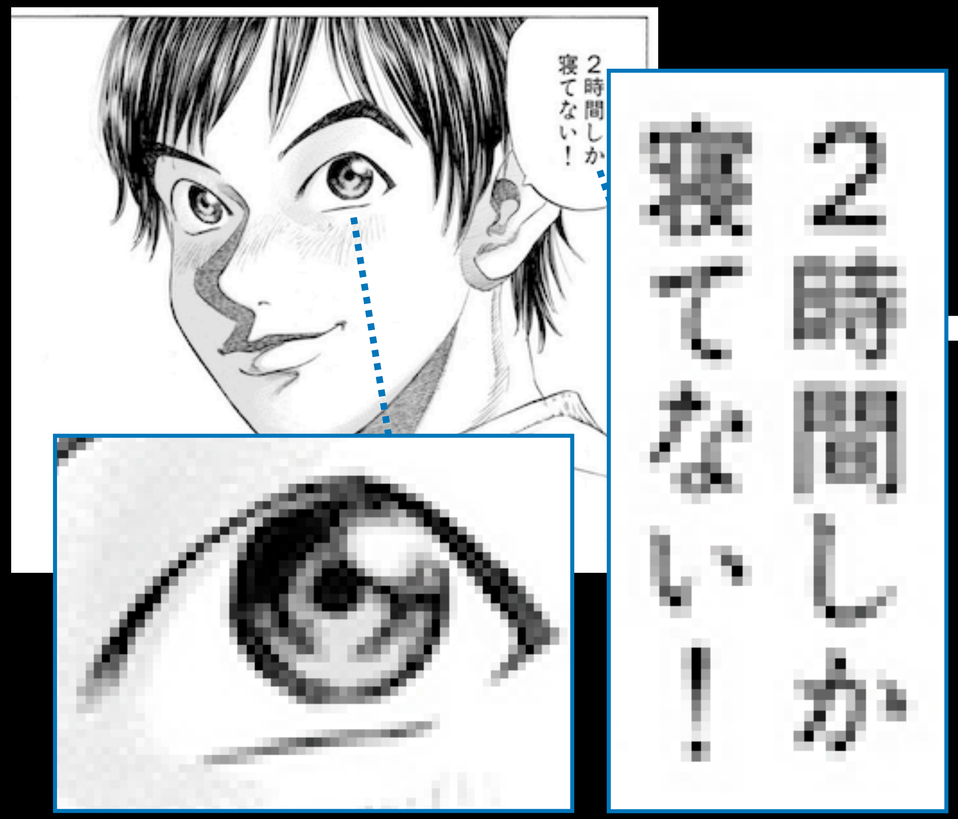


600x400

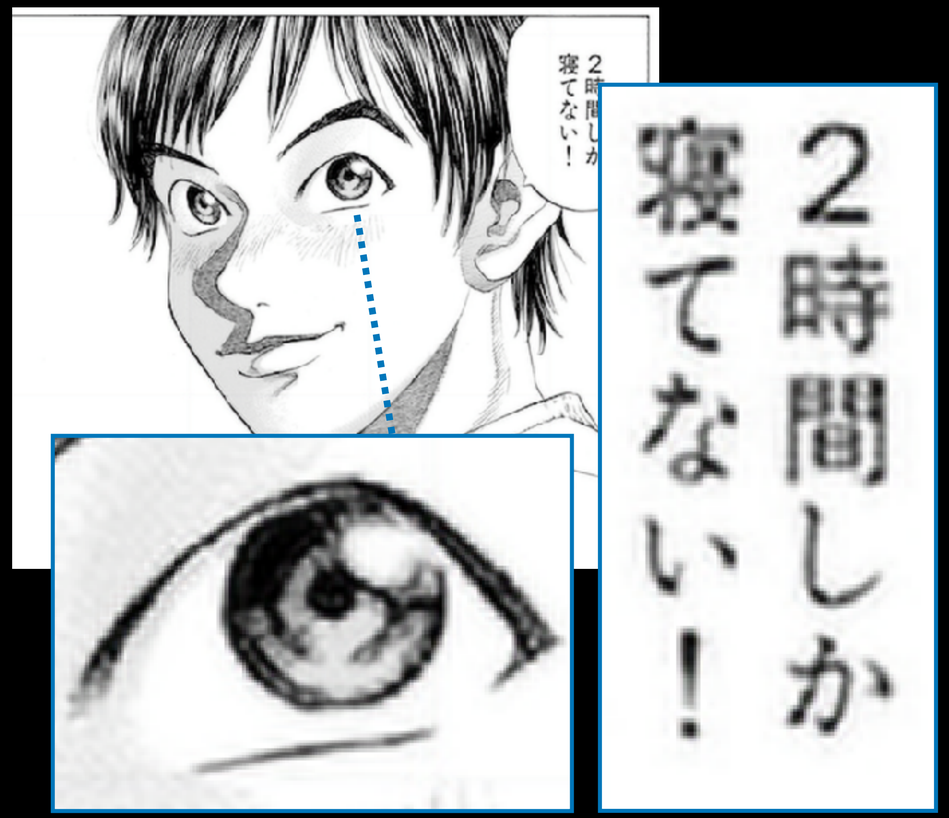
40 KB

ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

Super Resolution



Low Resolution

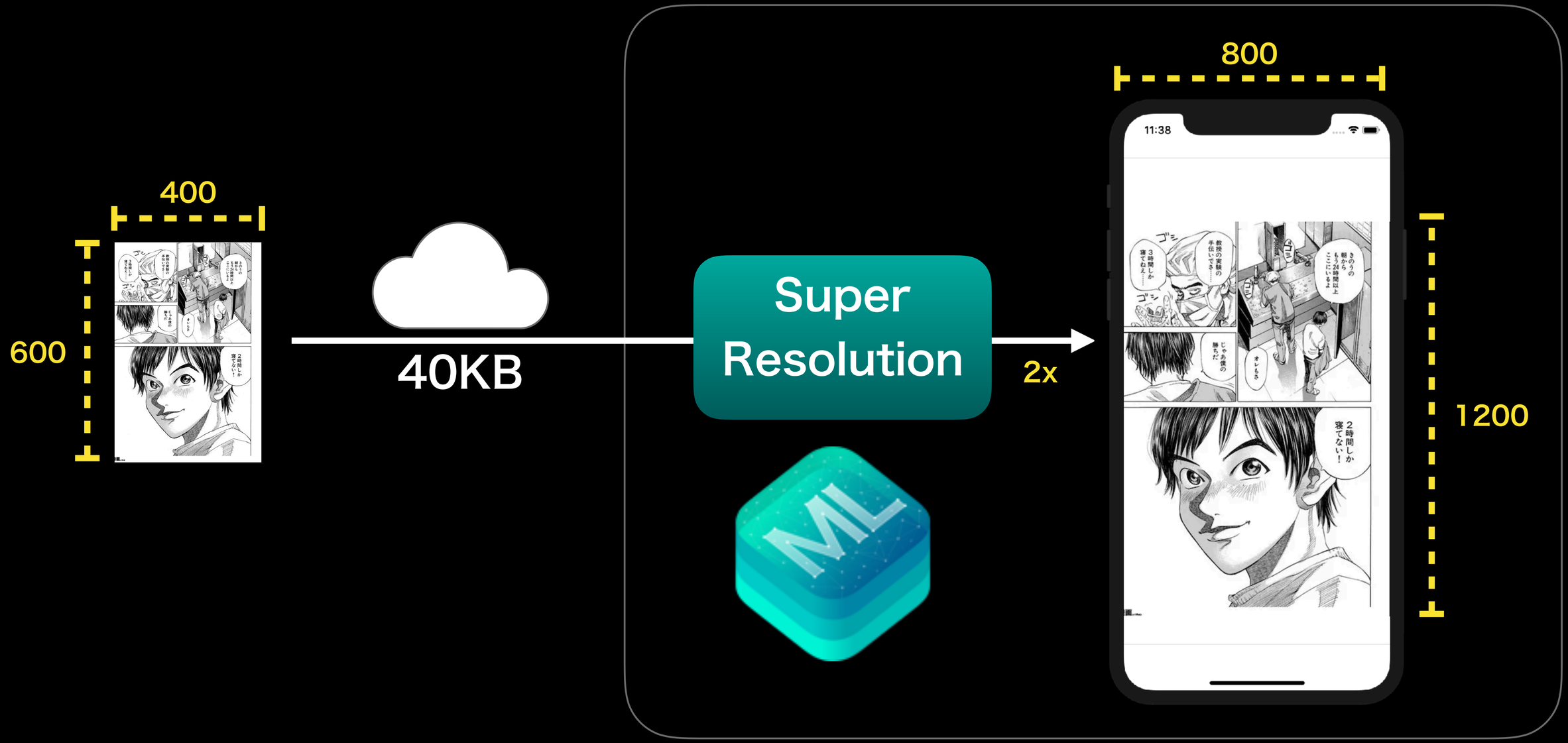


High Resolution

ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

CoreML

Super Resolution with CoreML



Demo

Super Resolution Mechanism

Image Super-Resolution Using Deep Convolutional Networks

Chao Dong et al. 2015

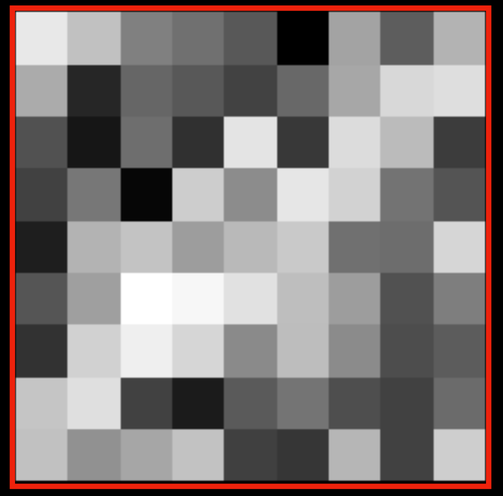
SRCNN

Super Resolution using Convolutional Neural Network

SRCNN

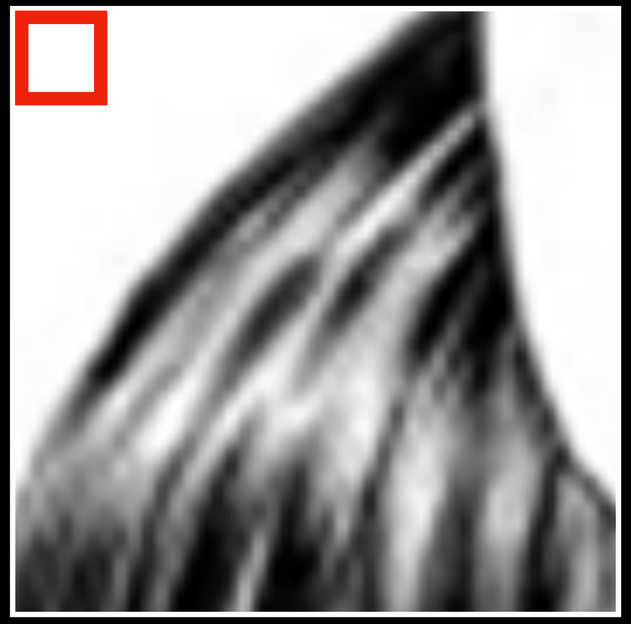
Super Resolution using Convolutional Neural Network

Kernel and Convolution

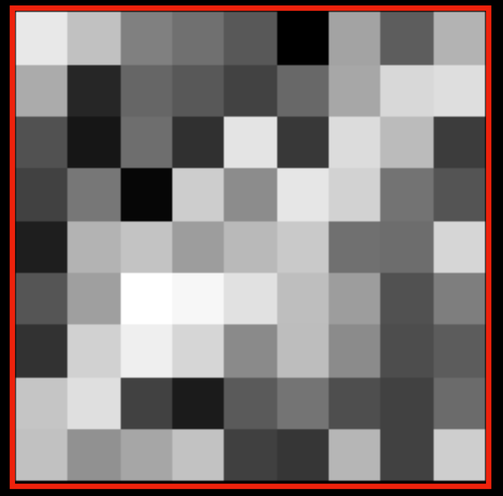


Kernel

Kernel and Convolution

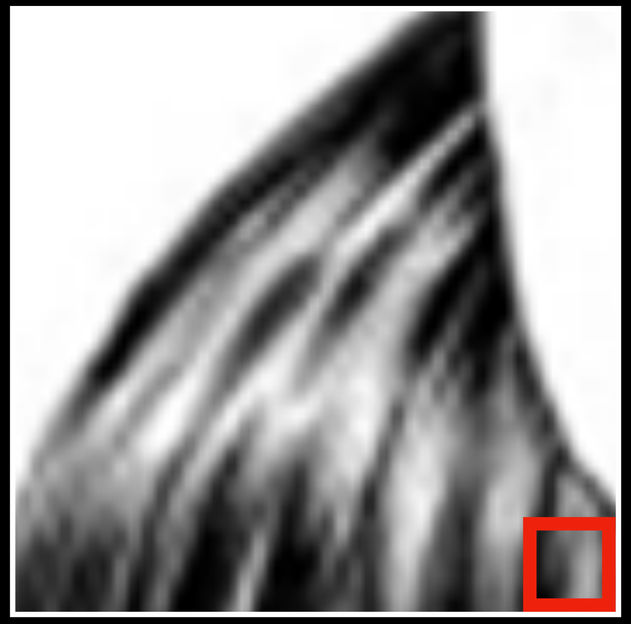


Input Image

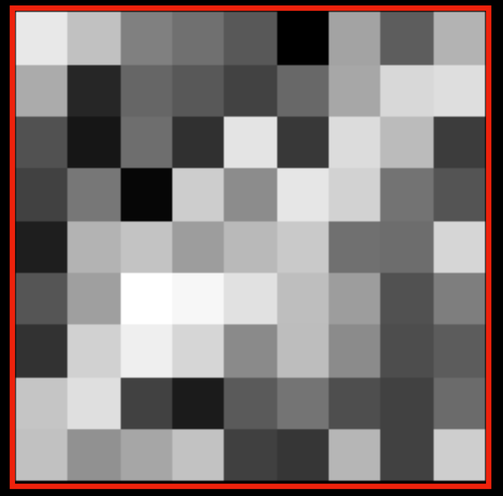


Kernel

Kernel and Convolution

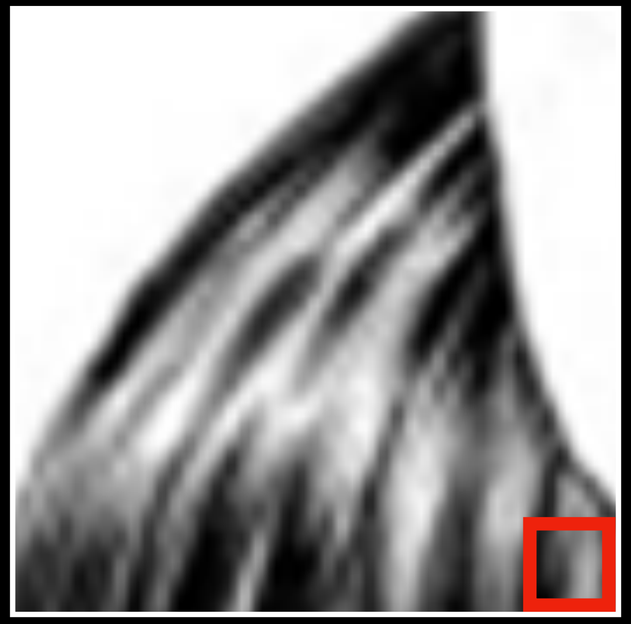


Input Image



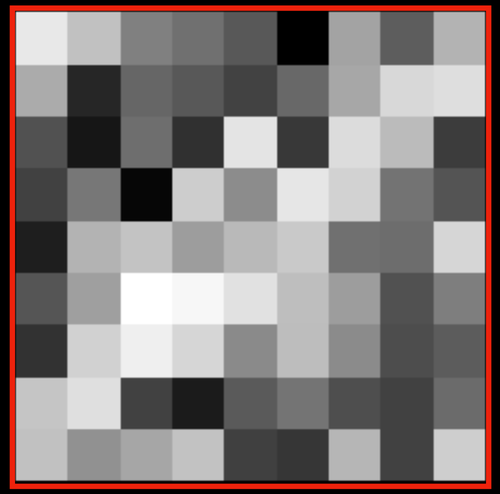
Kernel

Kernel and Convolution



Input Image

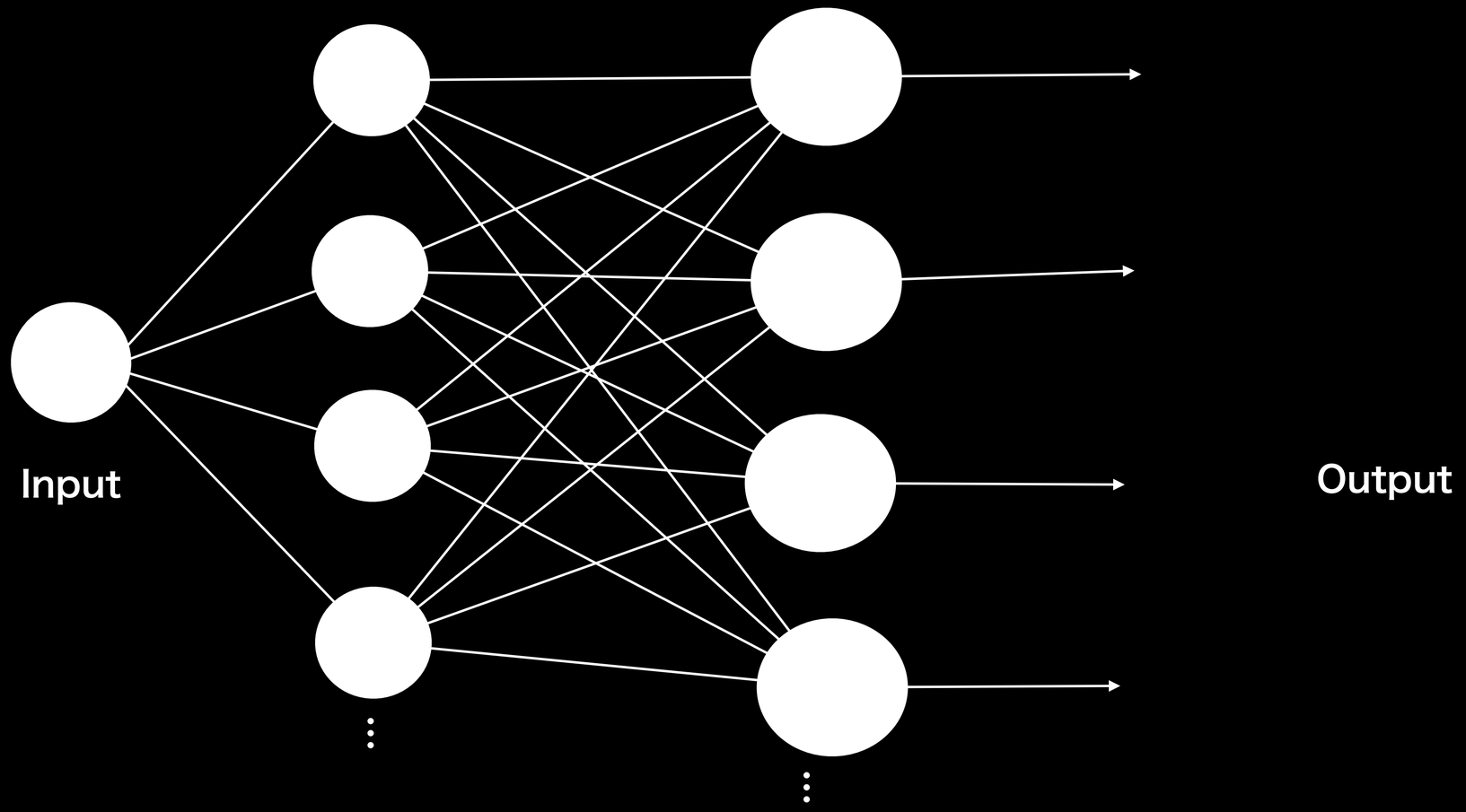
Convolution



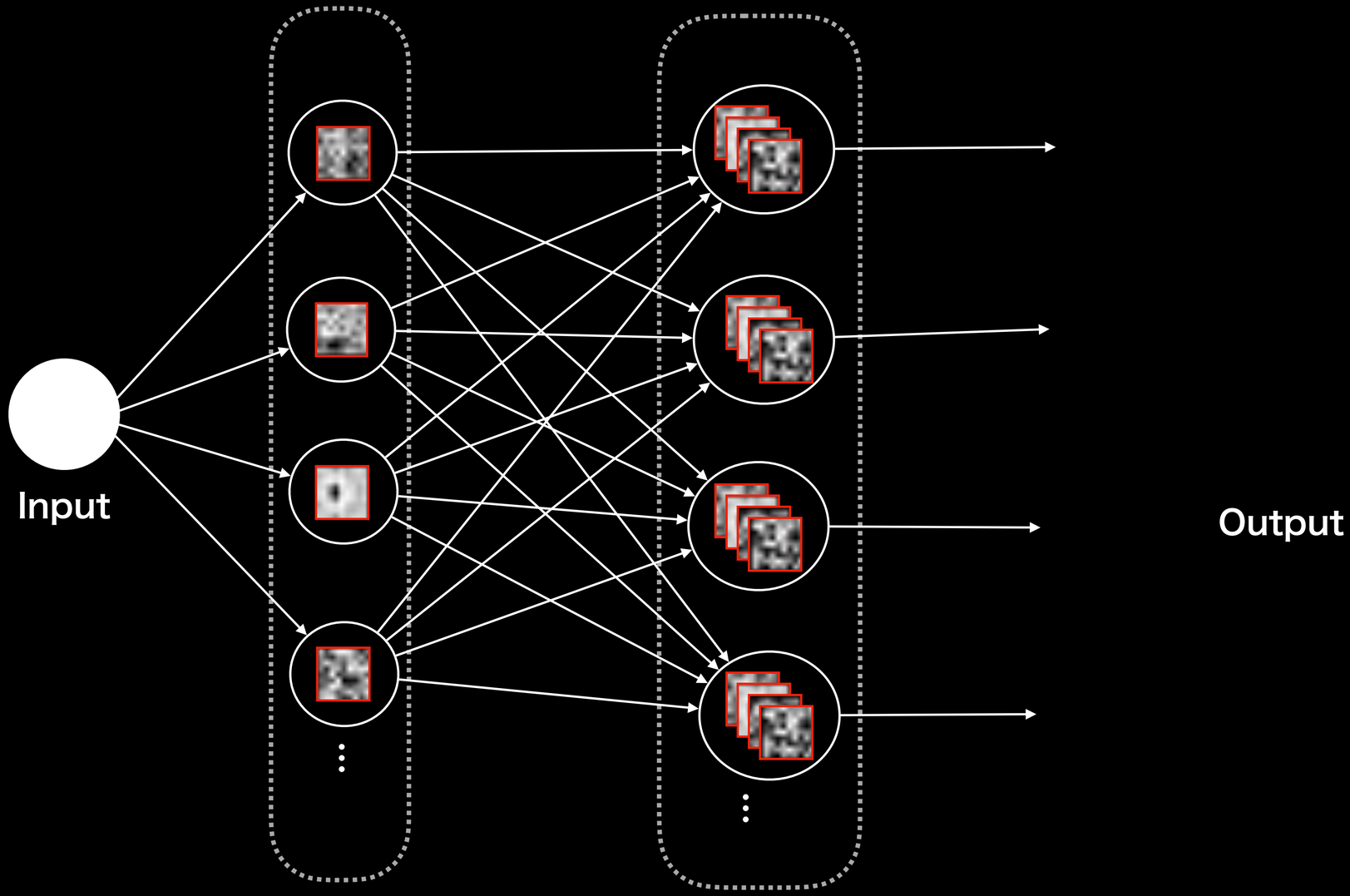
Kernel

Output Image

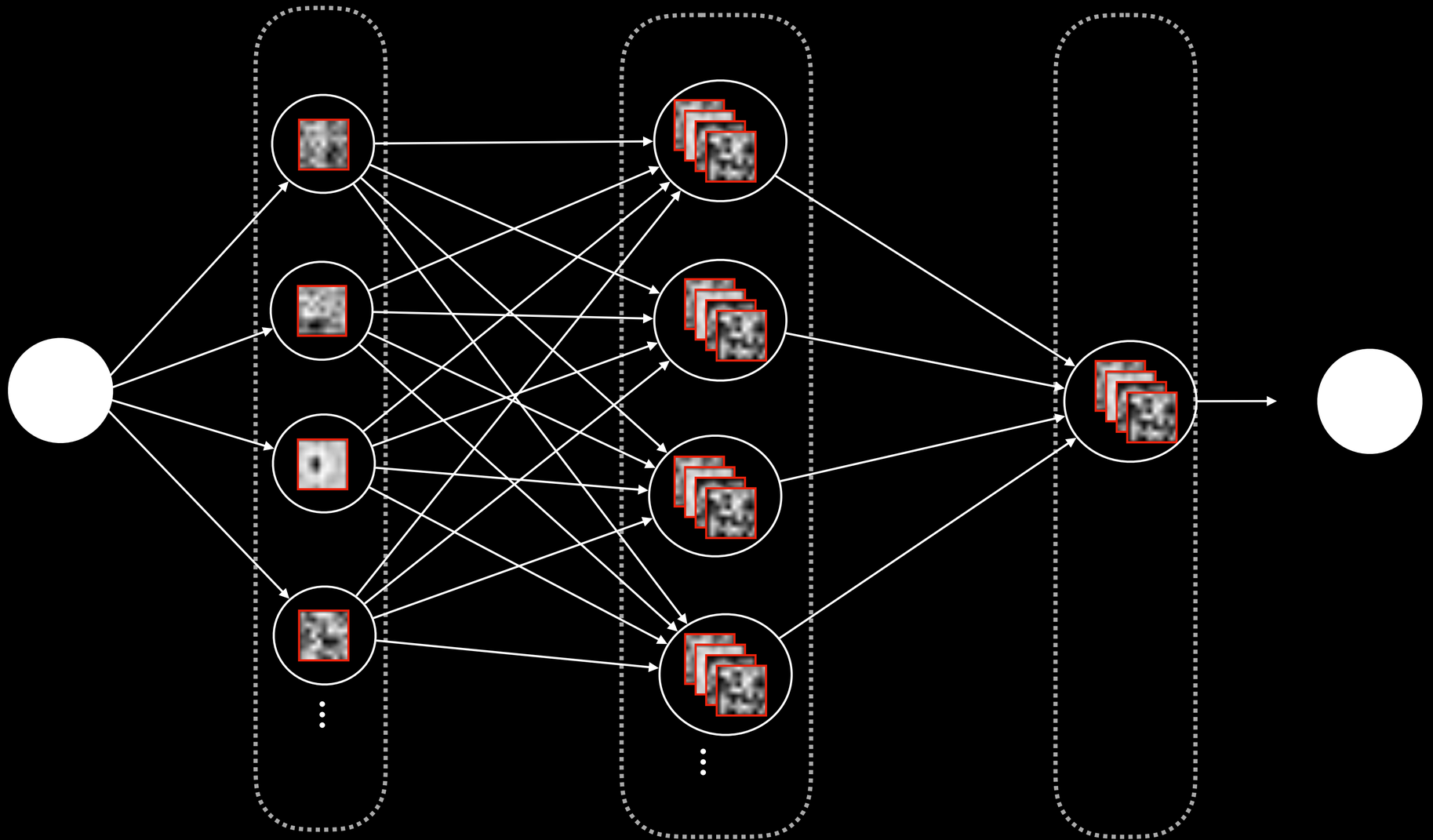
Neural Network



Convolutional Neural Network (CNN)

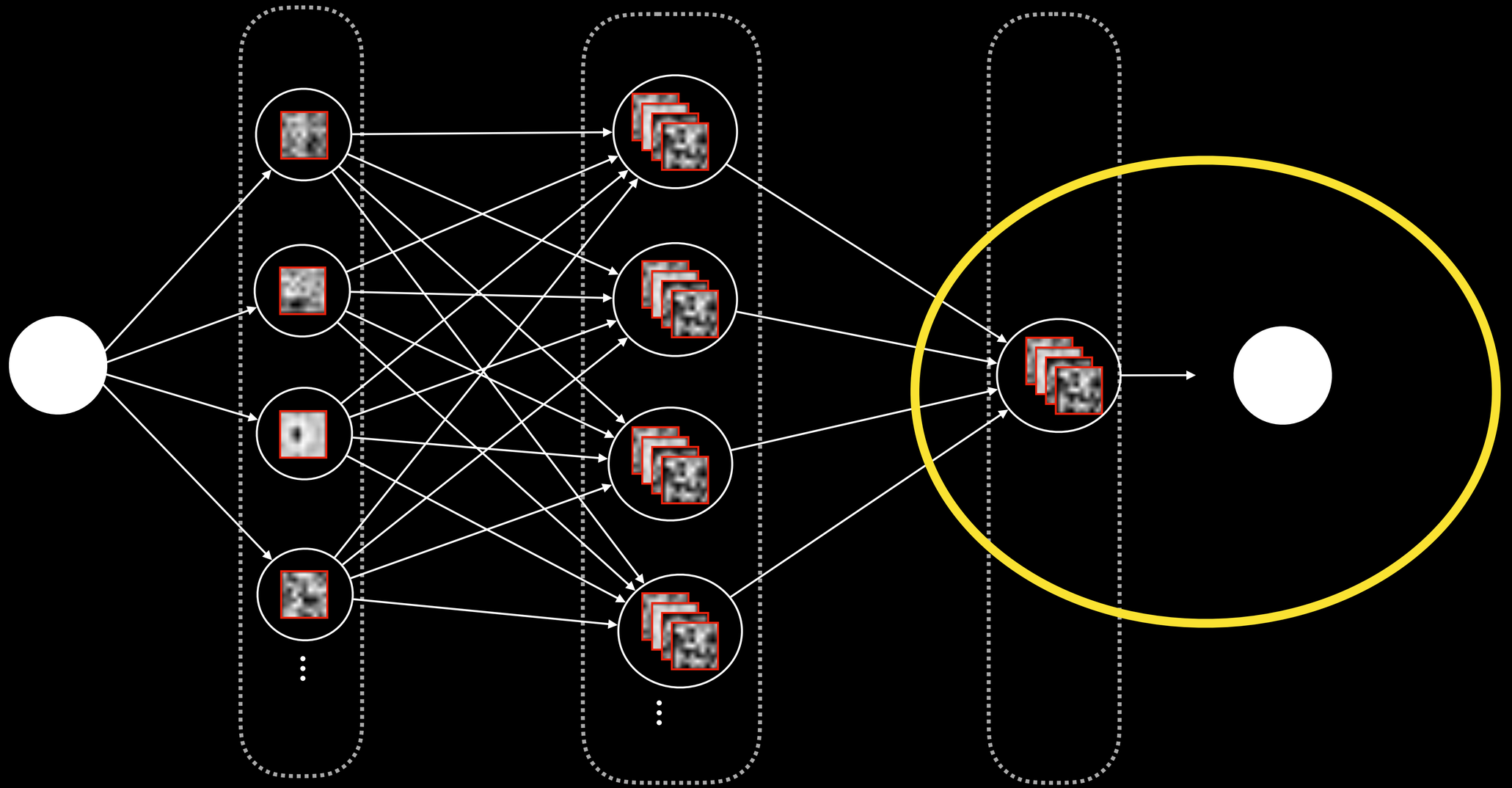


Super Resolution with CNN (SRCNN)



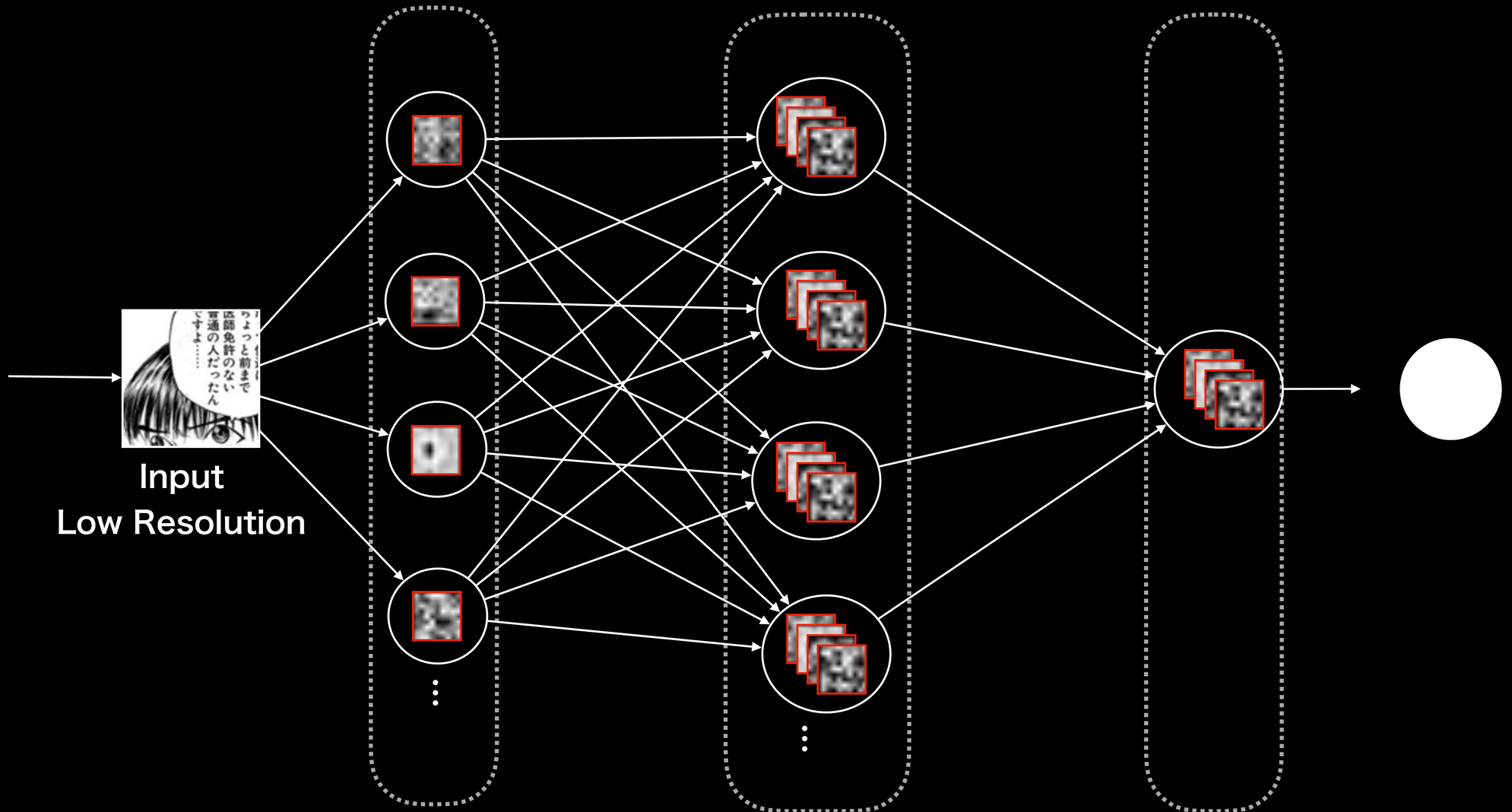
Conv Layer 1　　Conv Layer 2　　Conv Layer 3

Super Resolution with CNN (SRCNN)



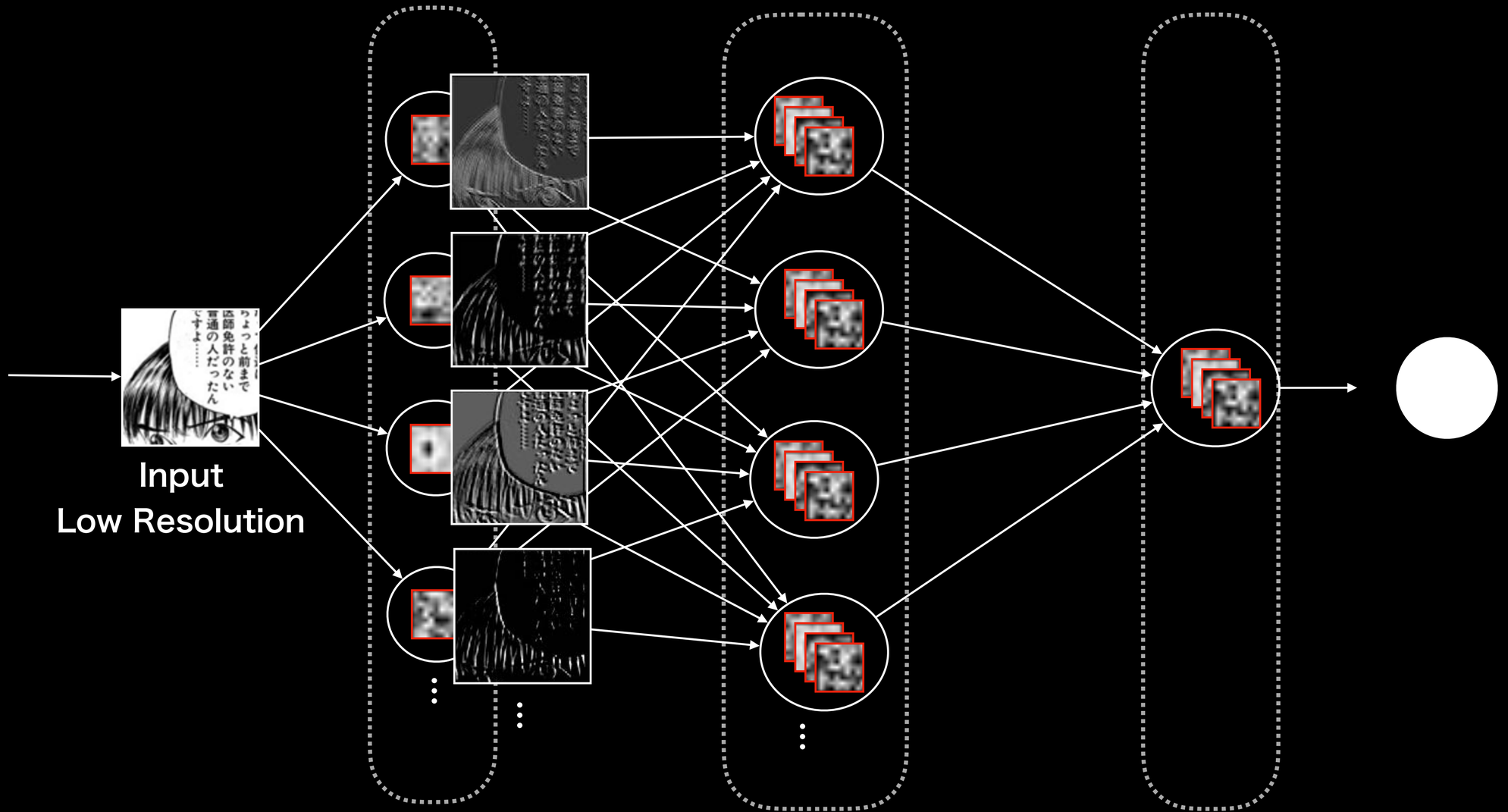
Conv Layer 1 Conv Layer 2 Conv Layer 3

Super Resolution with CNN (SRCNN)



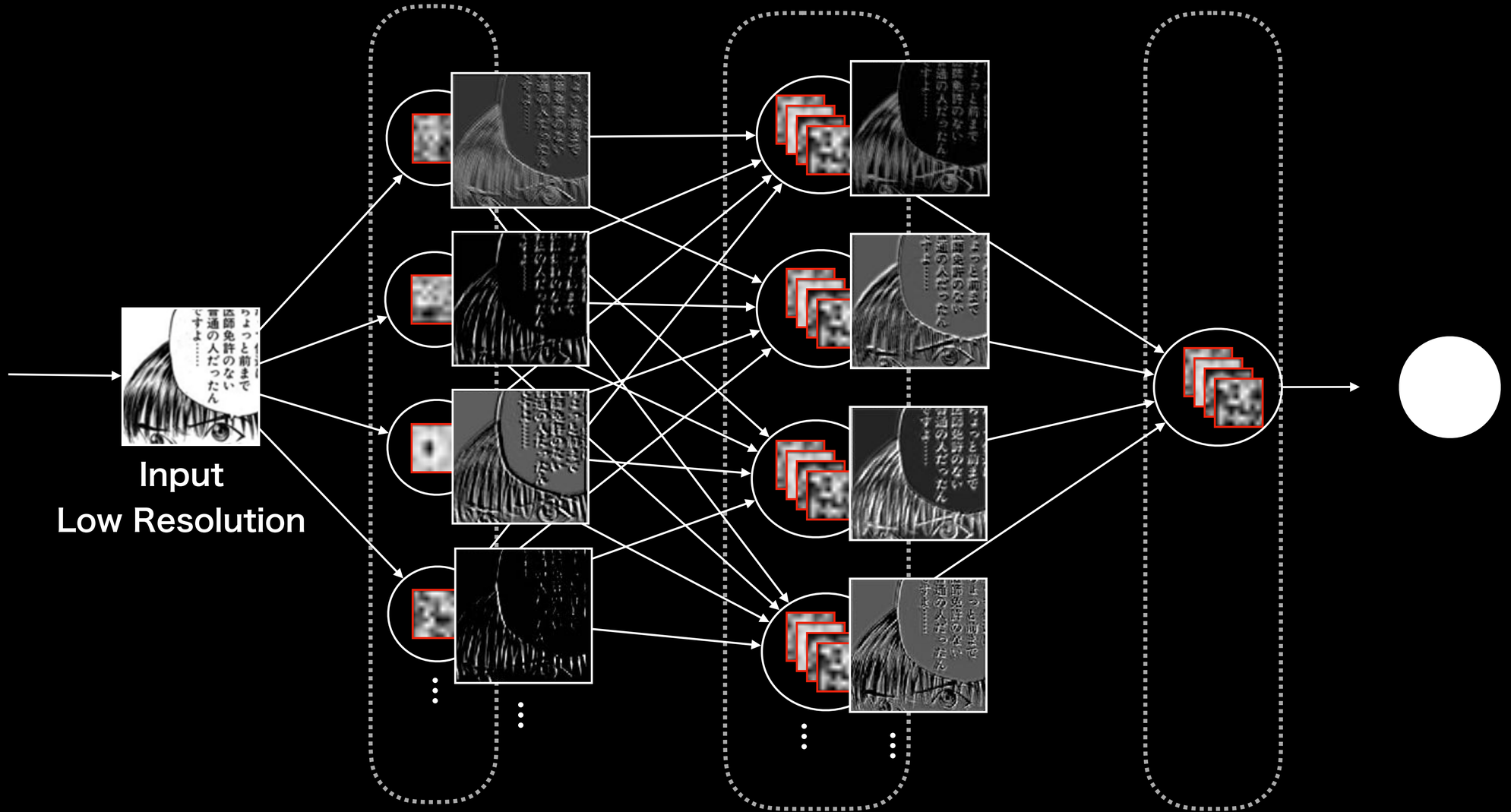
ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

Super Resolution with CNN (SRCNN)

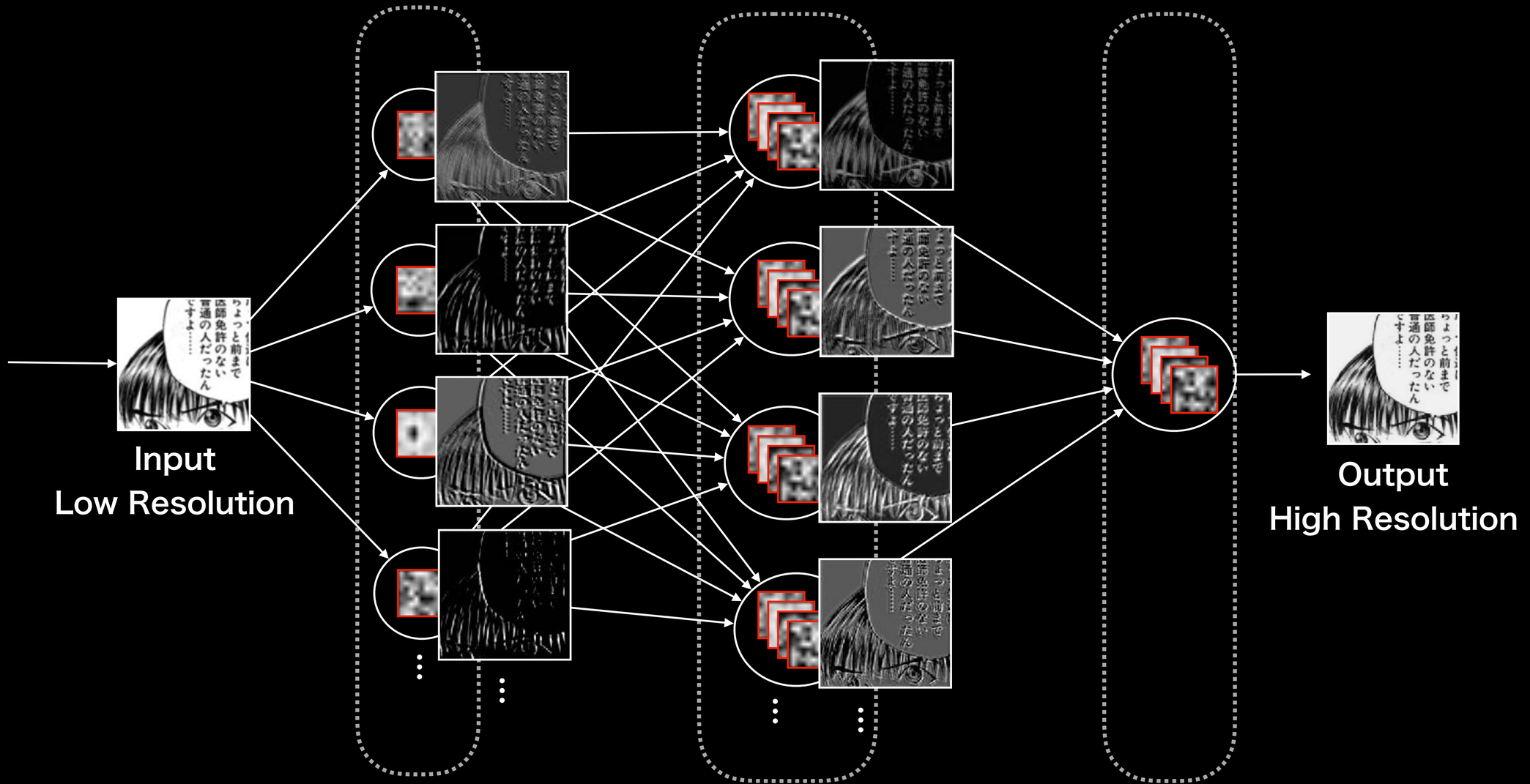


ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

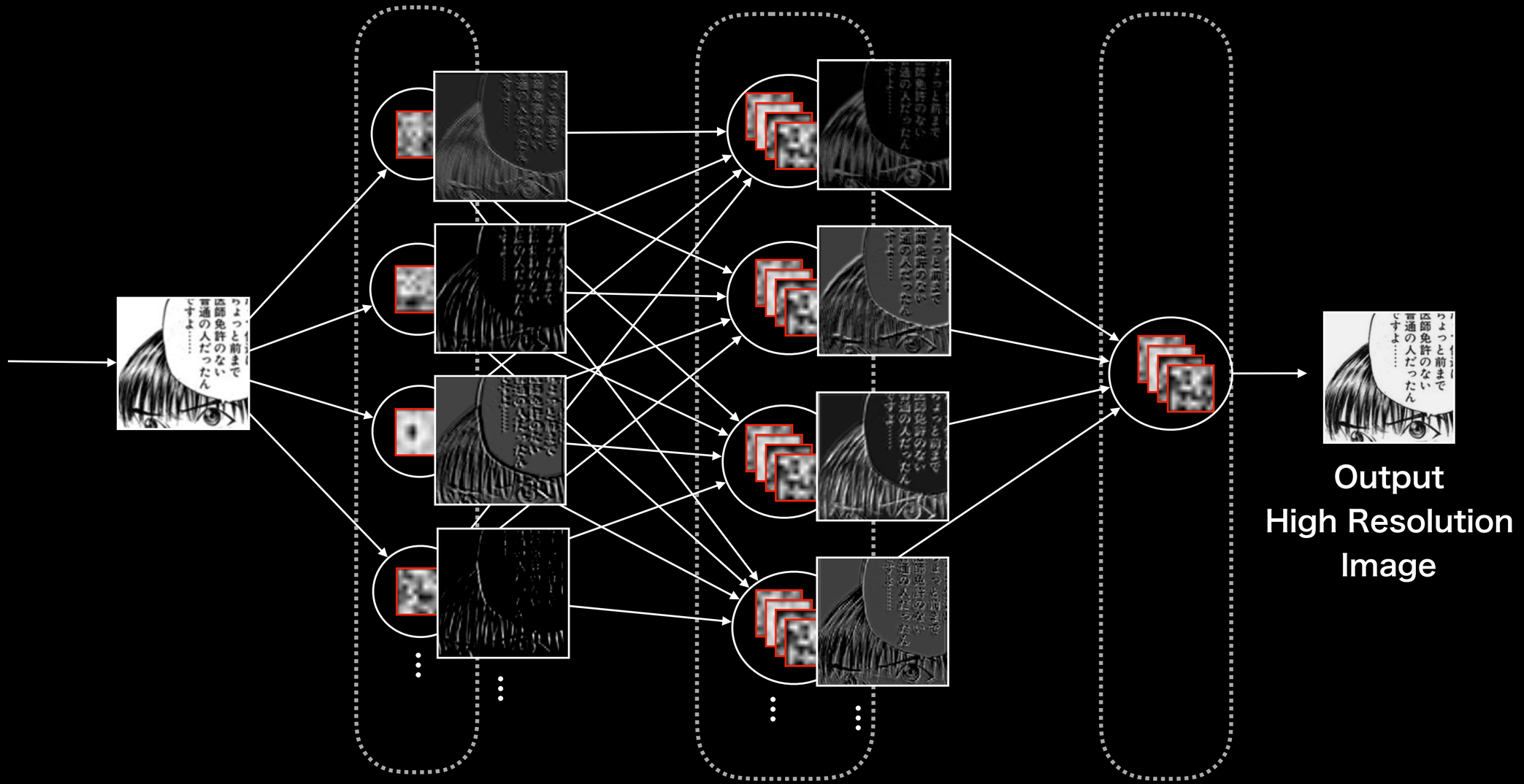
Super Resolution with CNN (SRCNN)



Super Resolution with CNN (SRCNN)



Super Resolution with CNN (SRCNN)



Output
High Resolution
Image

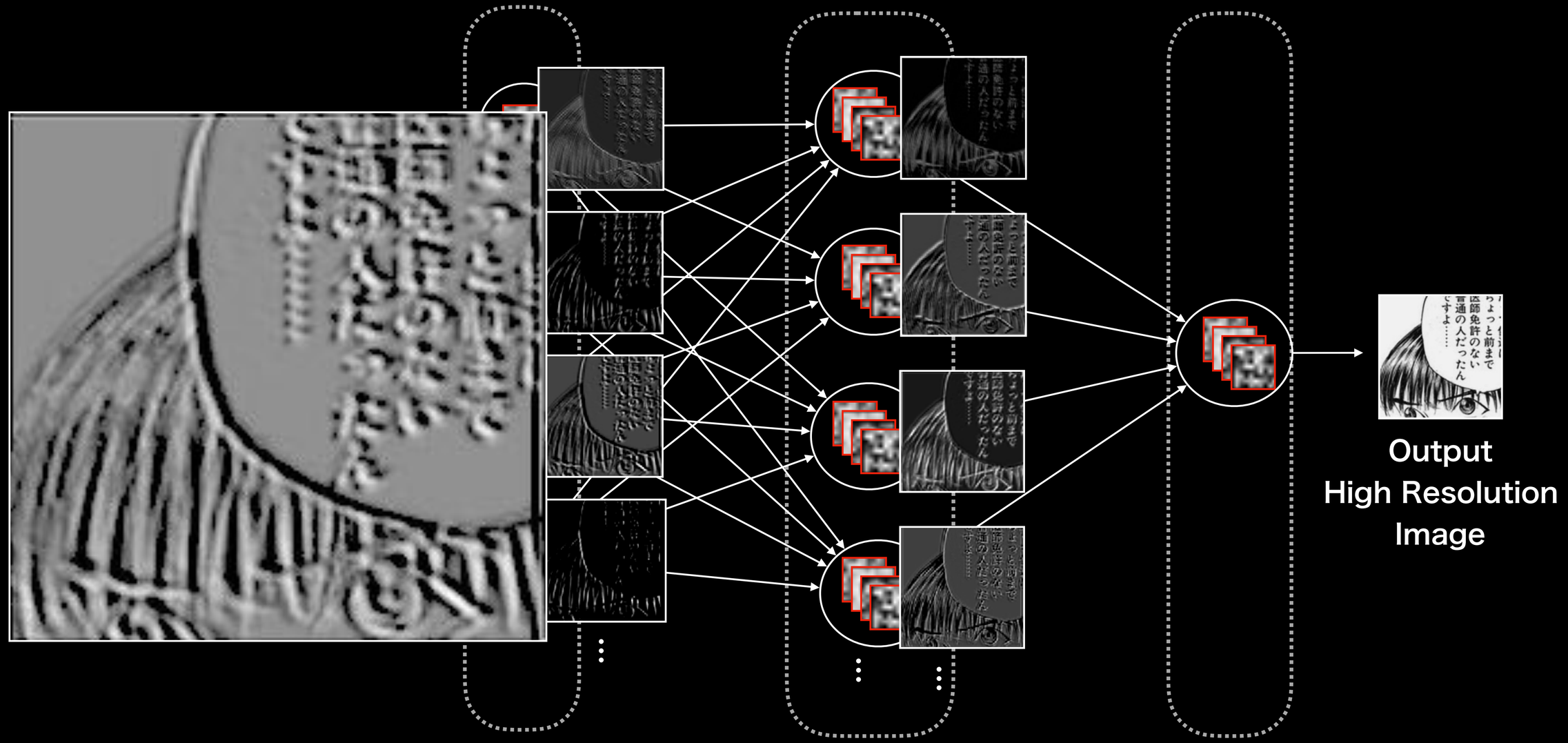
Conv Layer 1

Conv Layer 2

Conv Layer 3

ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

Super Resolution with CNN (SRCNN)



Output
High Resolution
Image

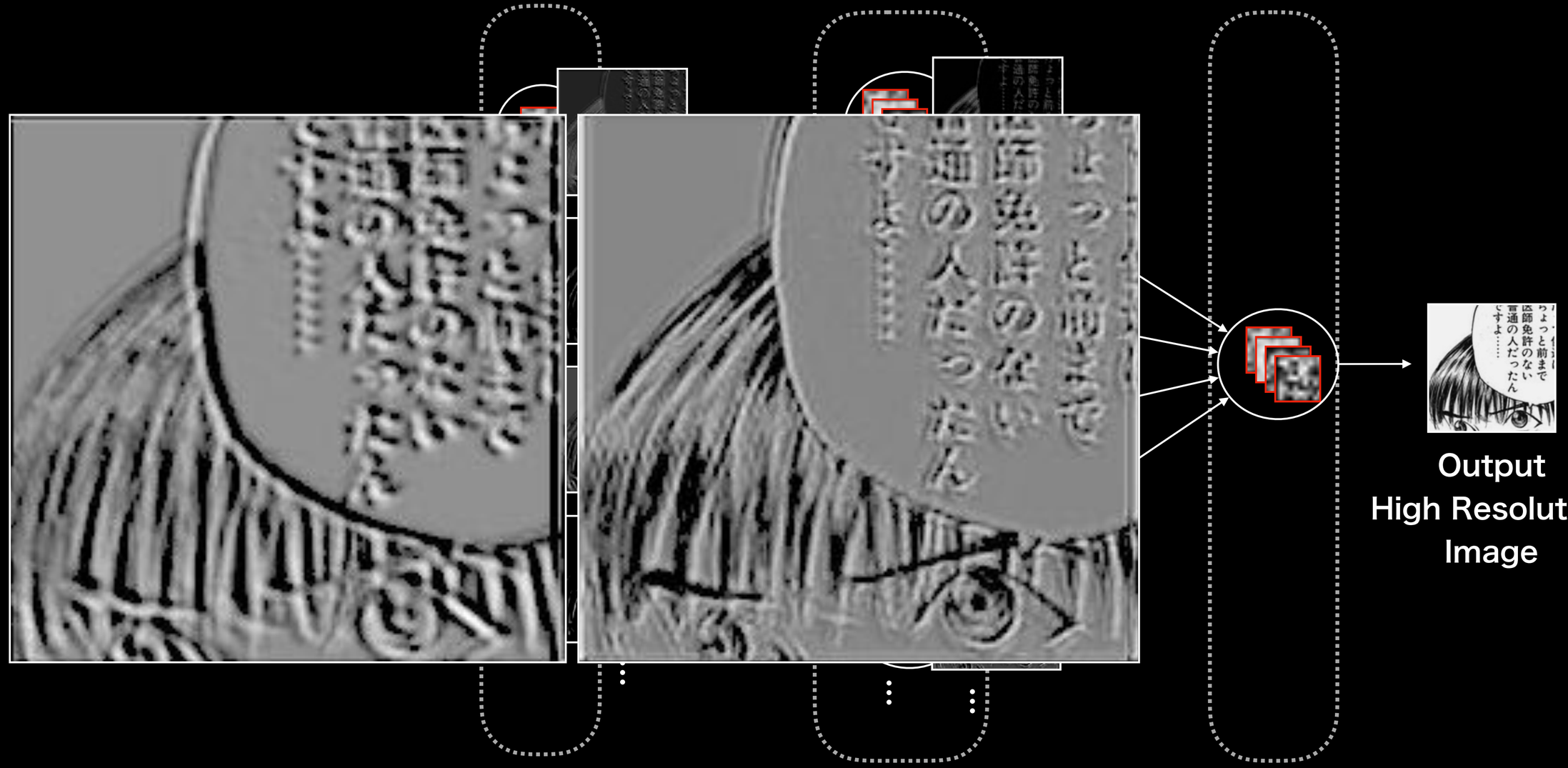
Conv Layer 1

Conv Layer 2

Conv Layer 3

ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

Super Resolution with CNN (SRCNN)



Output
High Resolution
Image

Conv Layer 1

Conv Layer 2

Conv Layer 3

Super Resolution with CNN (SRCNN)

Conv Layer 1

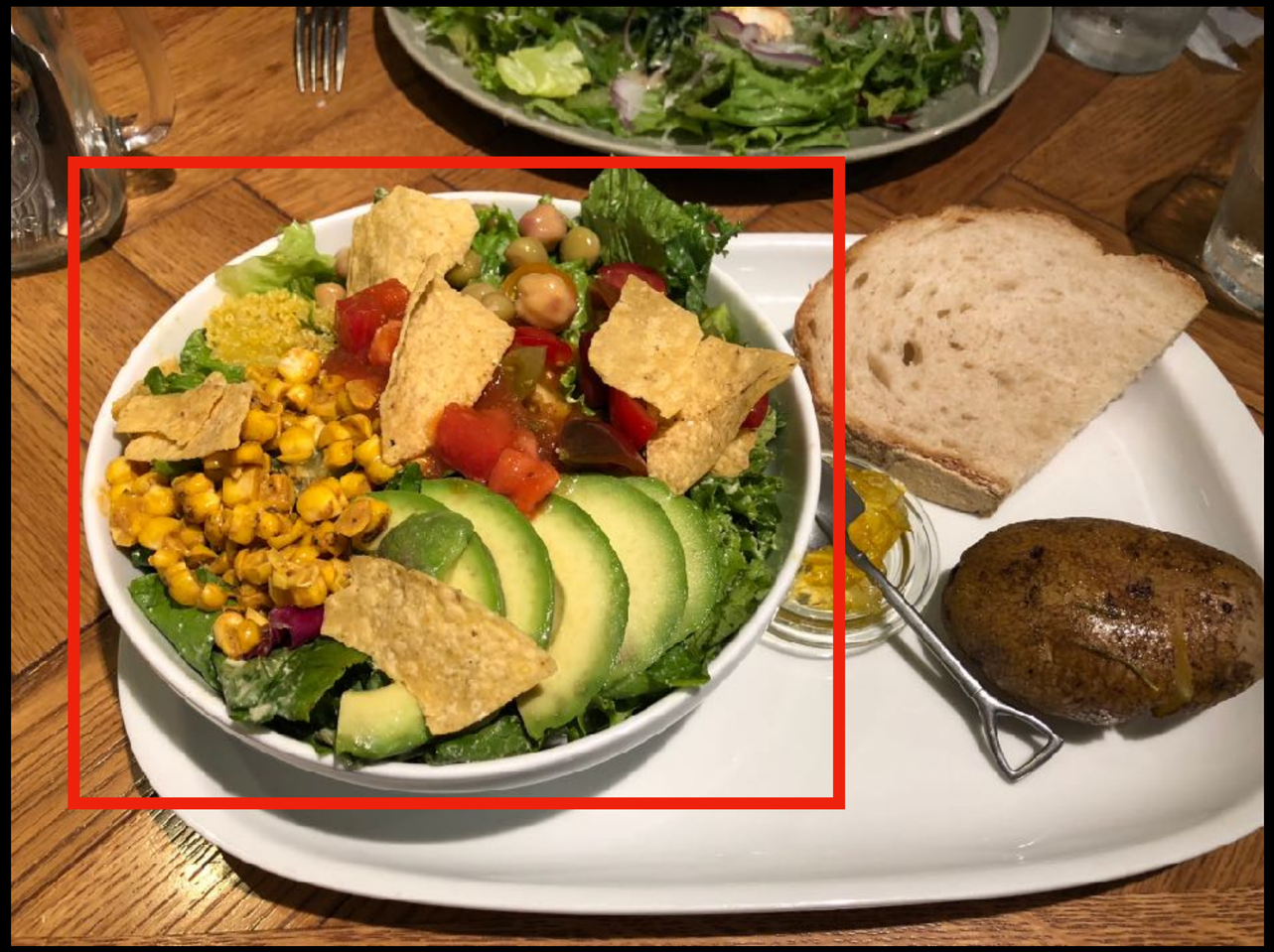
Conv Layer 2

Conv Layer 3

ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

Training

Hard to collect training data?

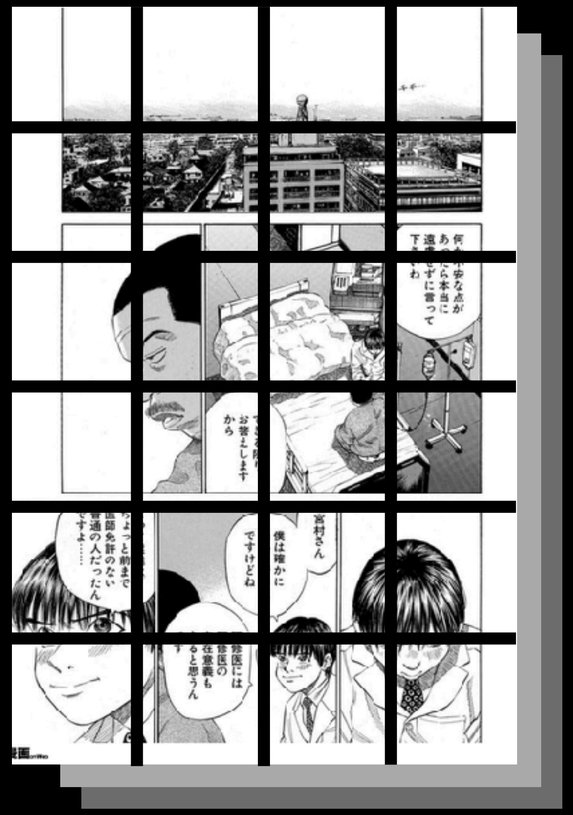


Easy to collect training data for SRCNN

HR Images

Easy to collect training data for SRCNN

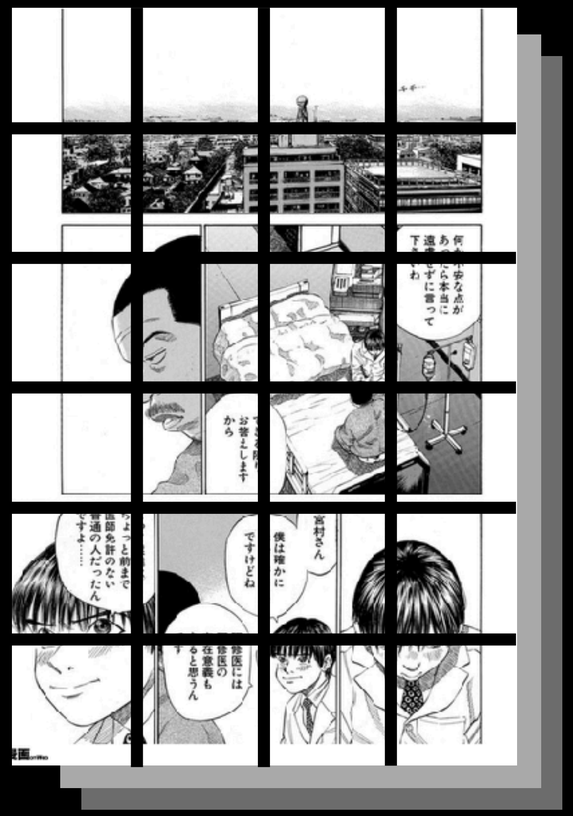
HR Images



ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

Easy to collect training data for SRCNN

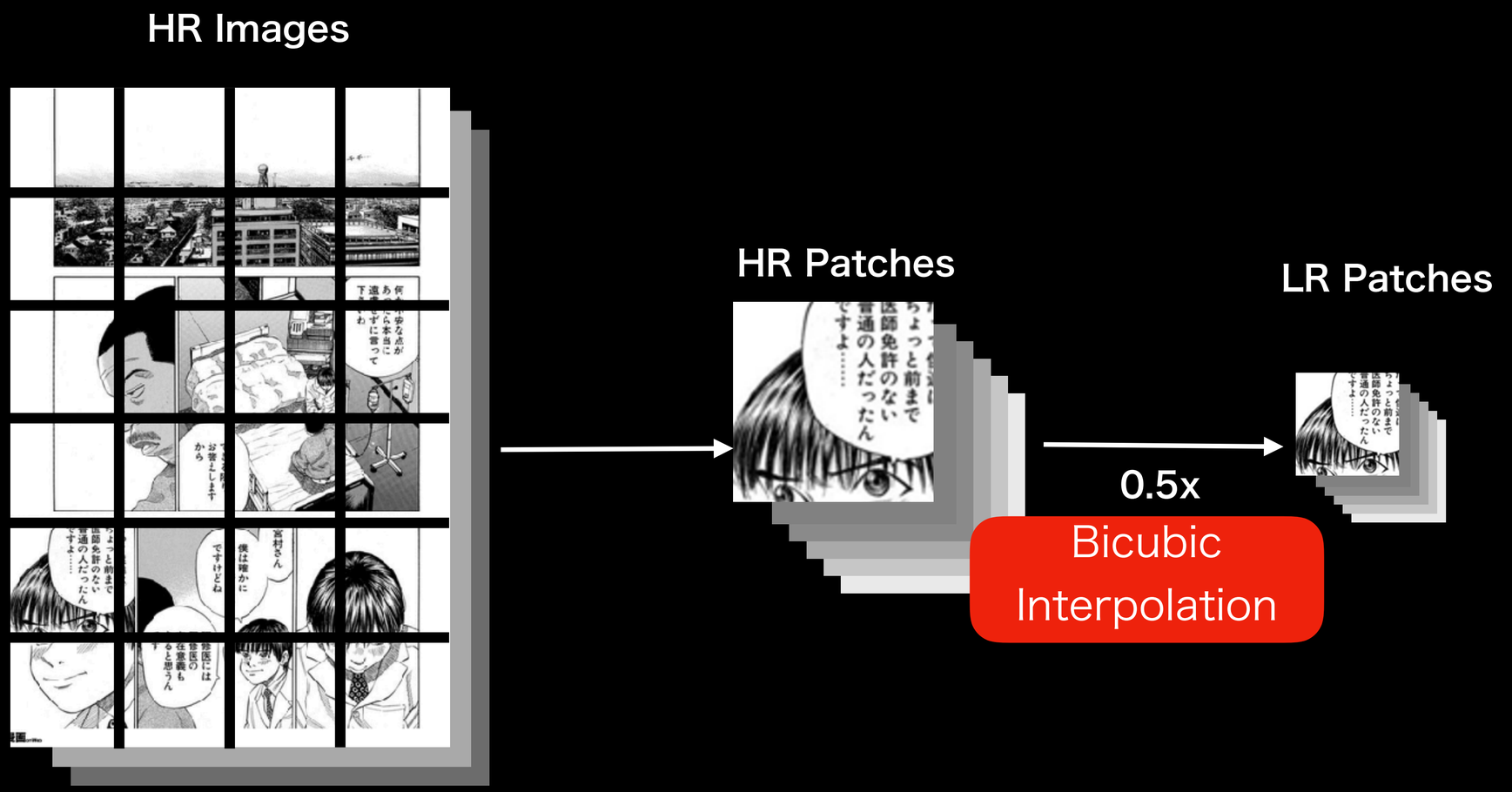
HR Images



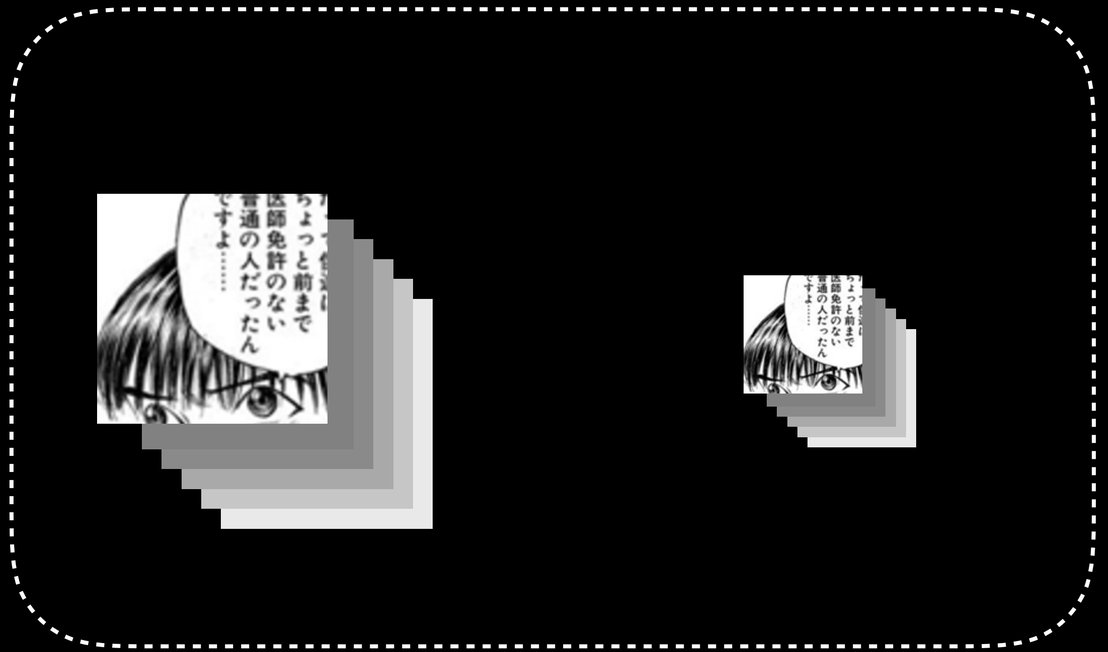
HR Patches

ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

Easy to collect training data for SRCNN



ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

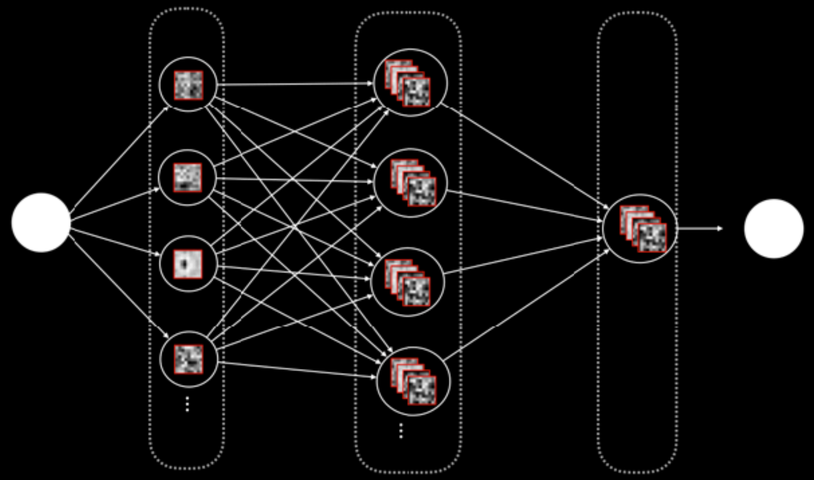


Training Set


ちょっと前まで
医師免許のない
普通の人だったん
ですよ……

Training SRCNN

LR Patch
Input

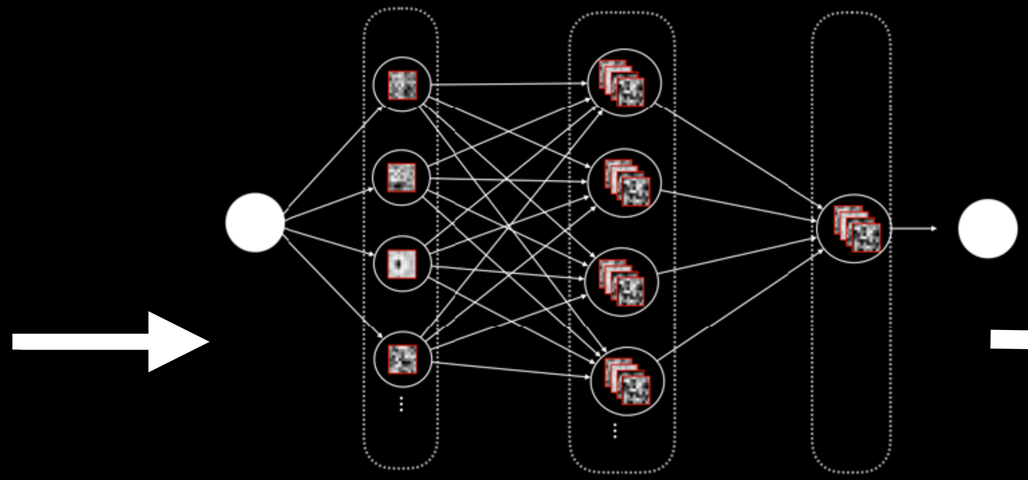


HR Patch

ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

Training SRCNN

LR Patch
Input

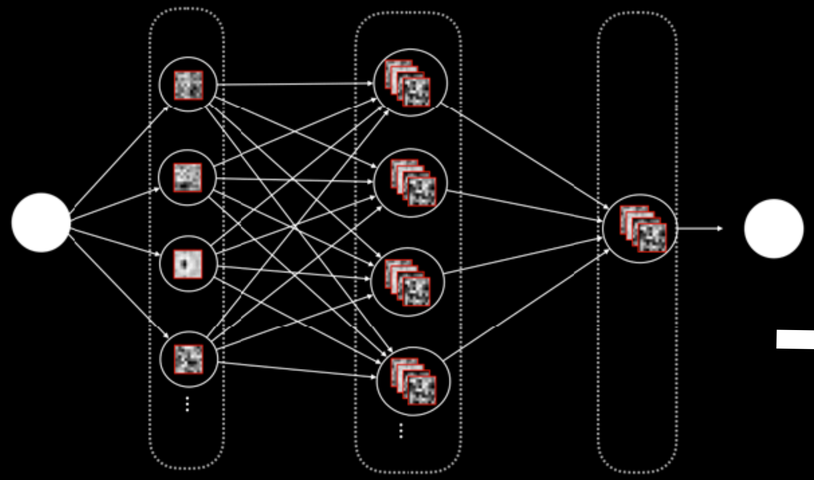


Output

HR Patch

Training SRCNN

LR Patch Input



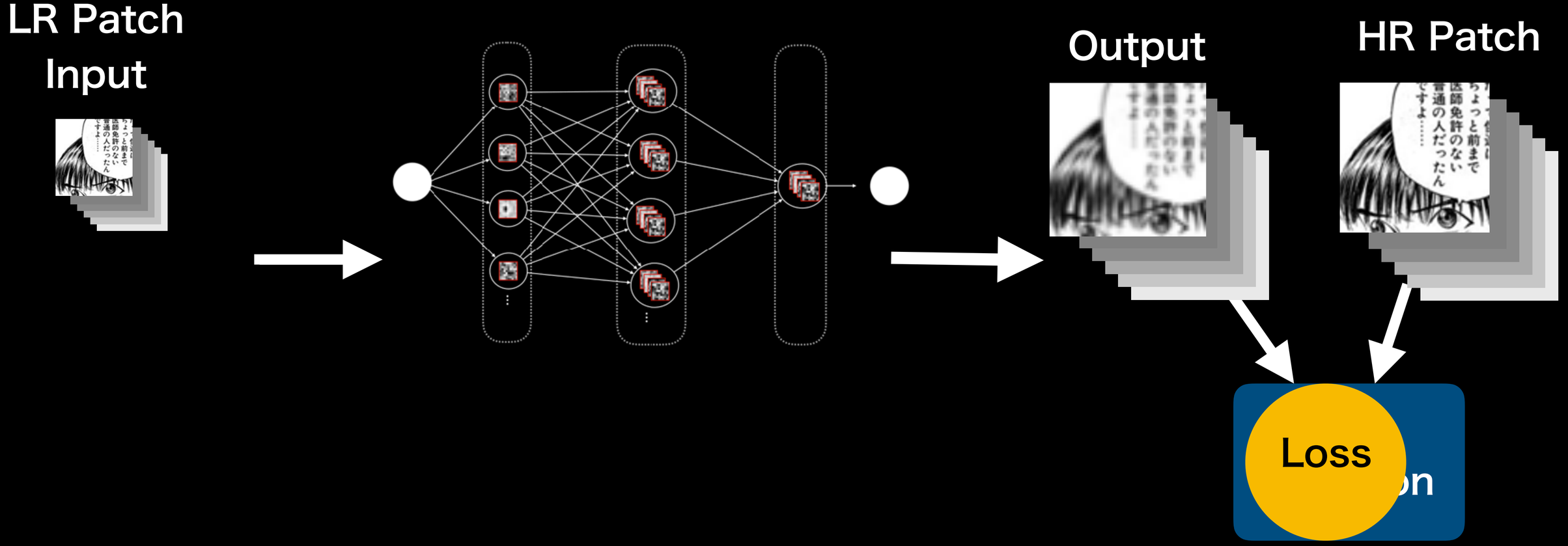
Output

HR Patch

Loss Function

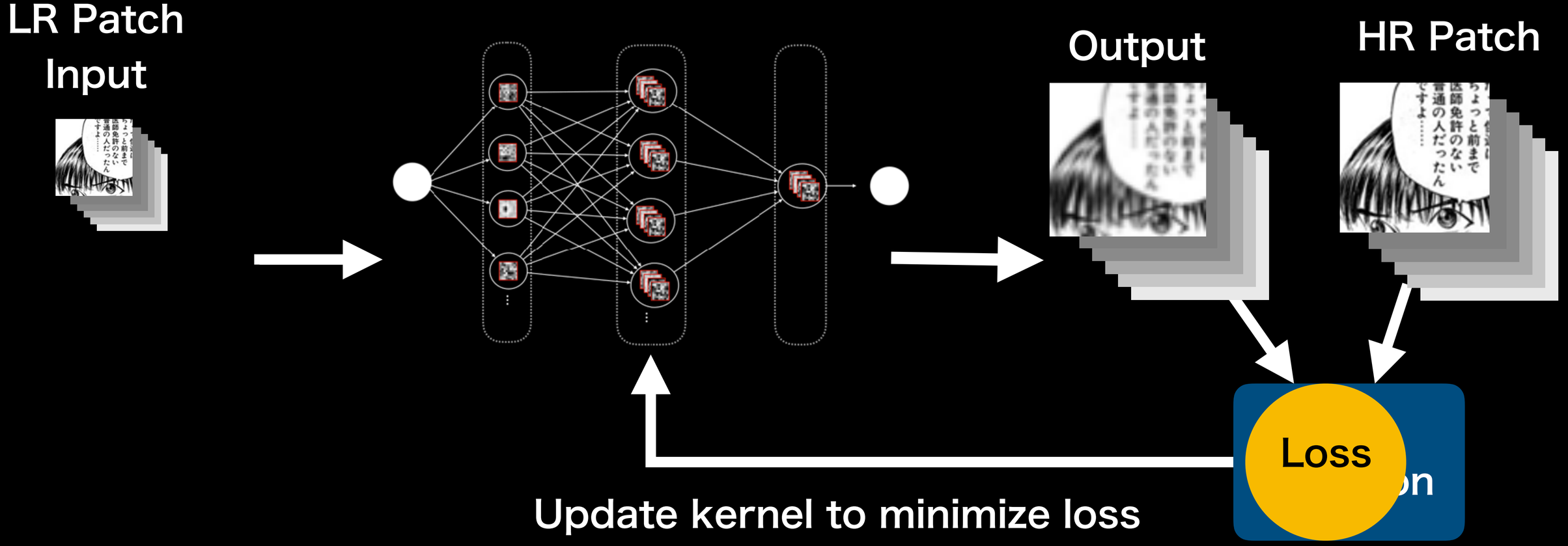
ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

Training SRCNN



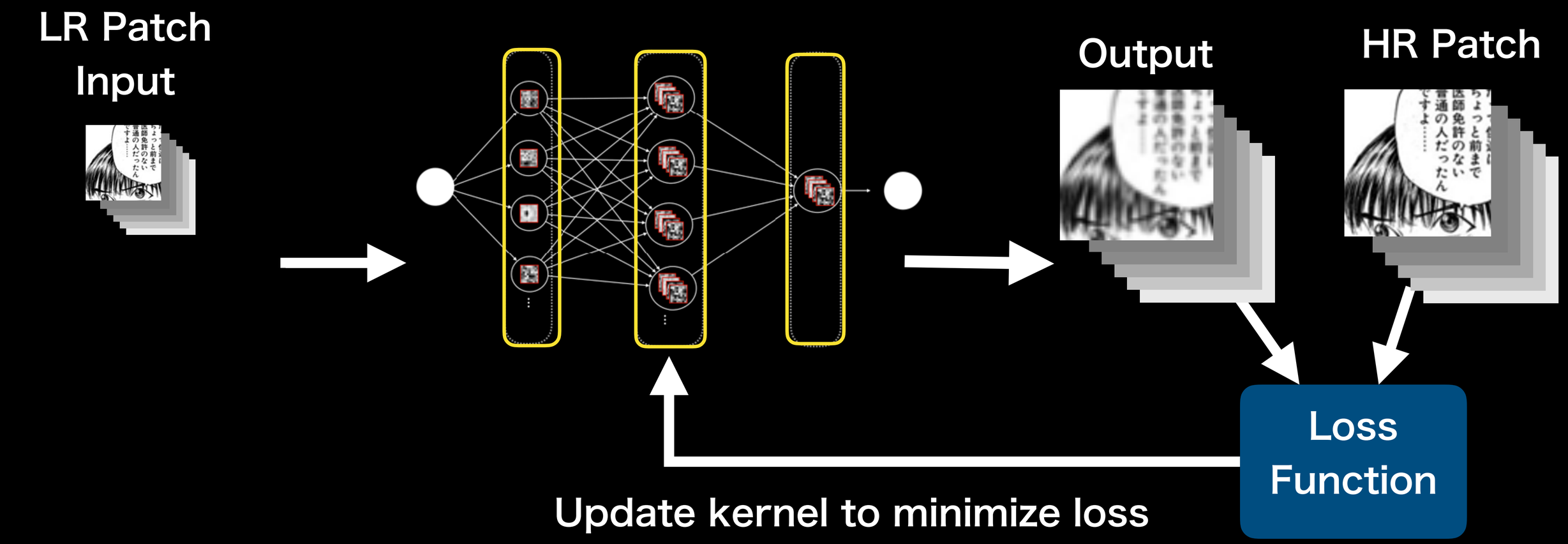
ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

Training SRCNN



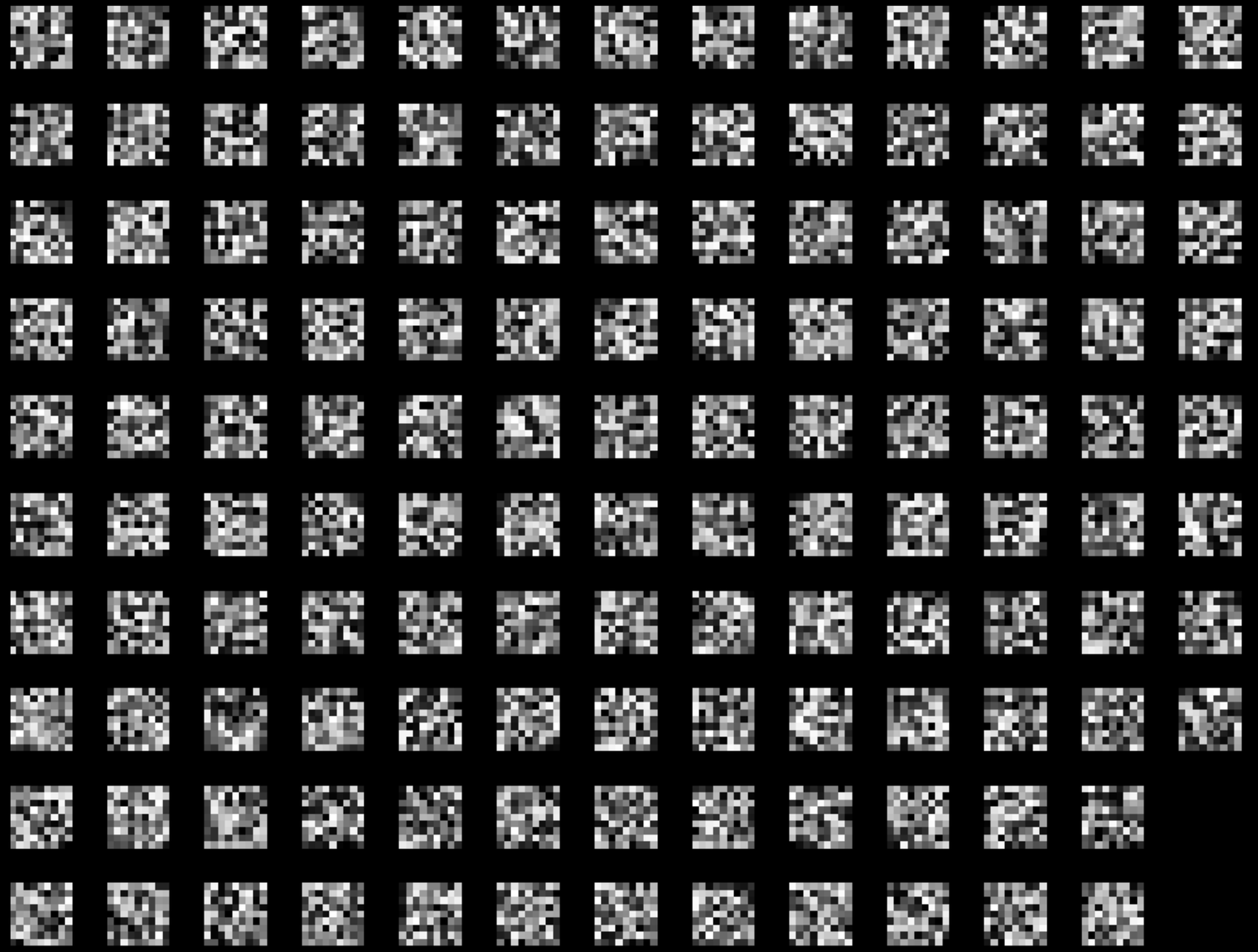
ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

Training SRCNN

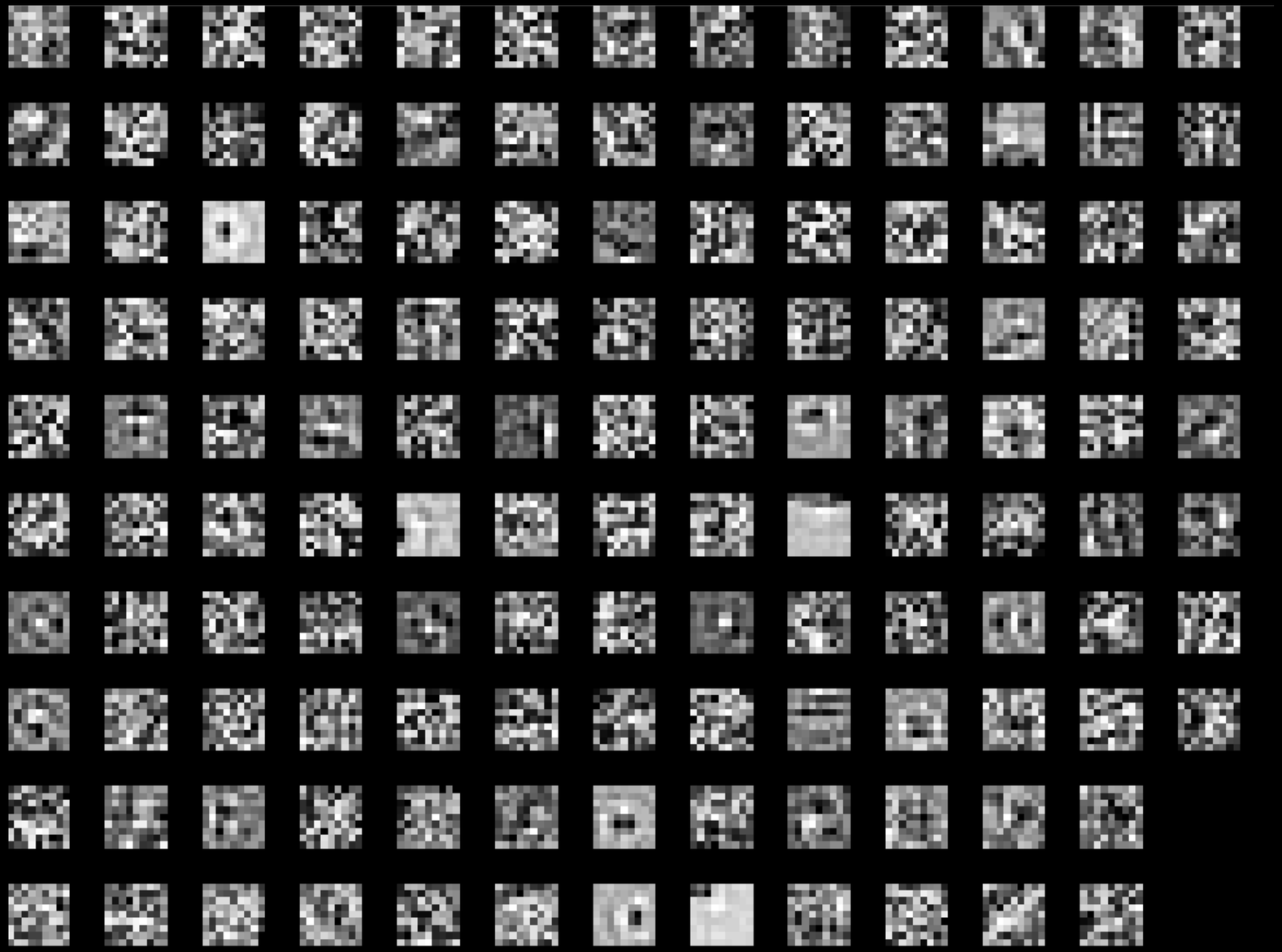


ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

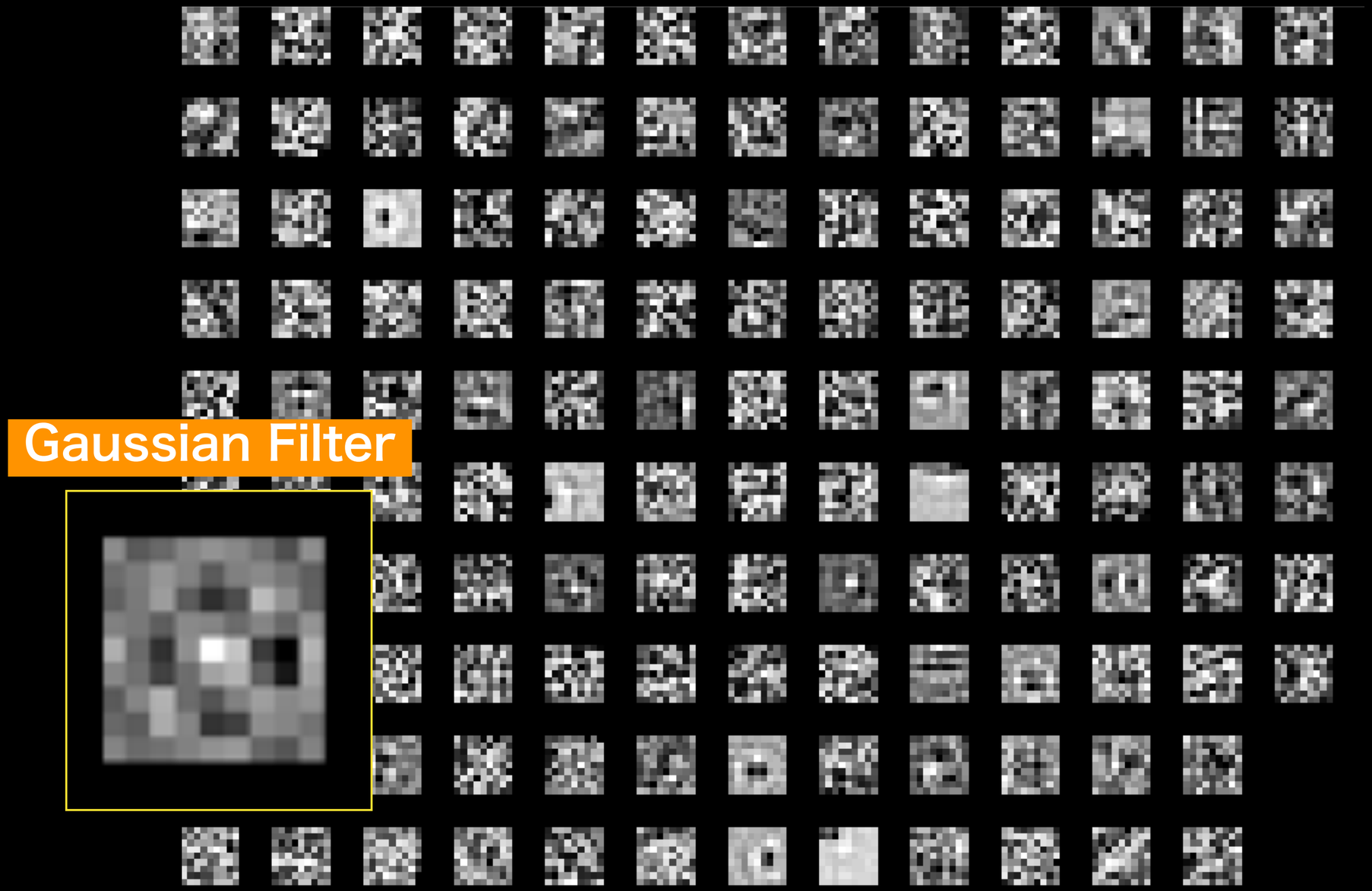
Kernel Weight (before training)



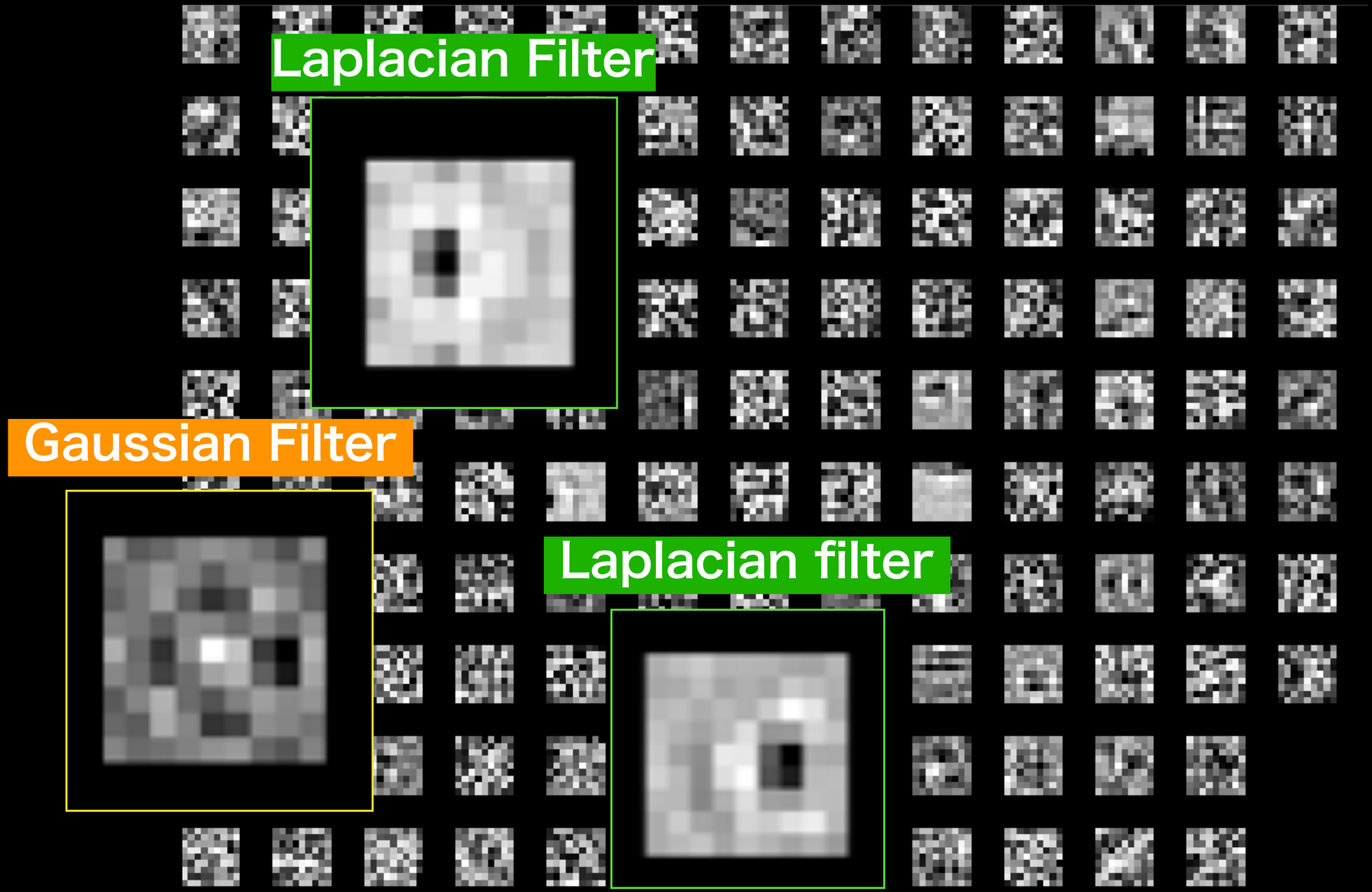
Kernel Weight (after training)



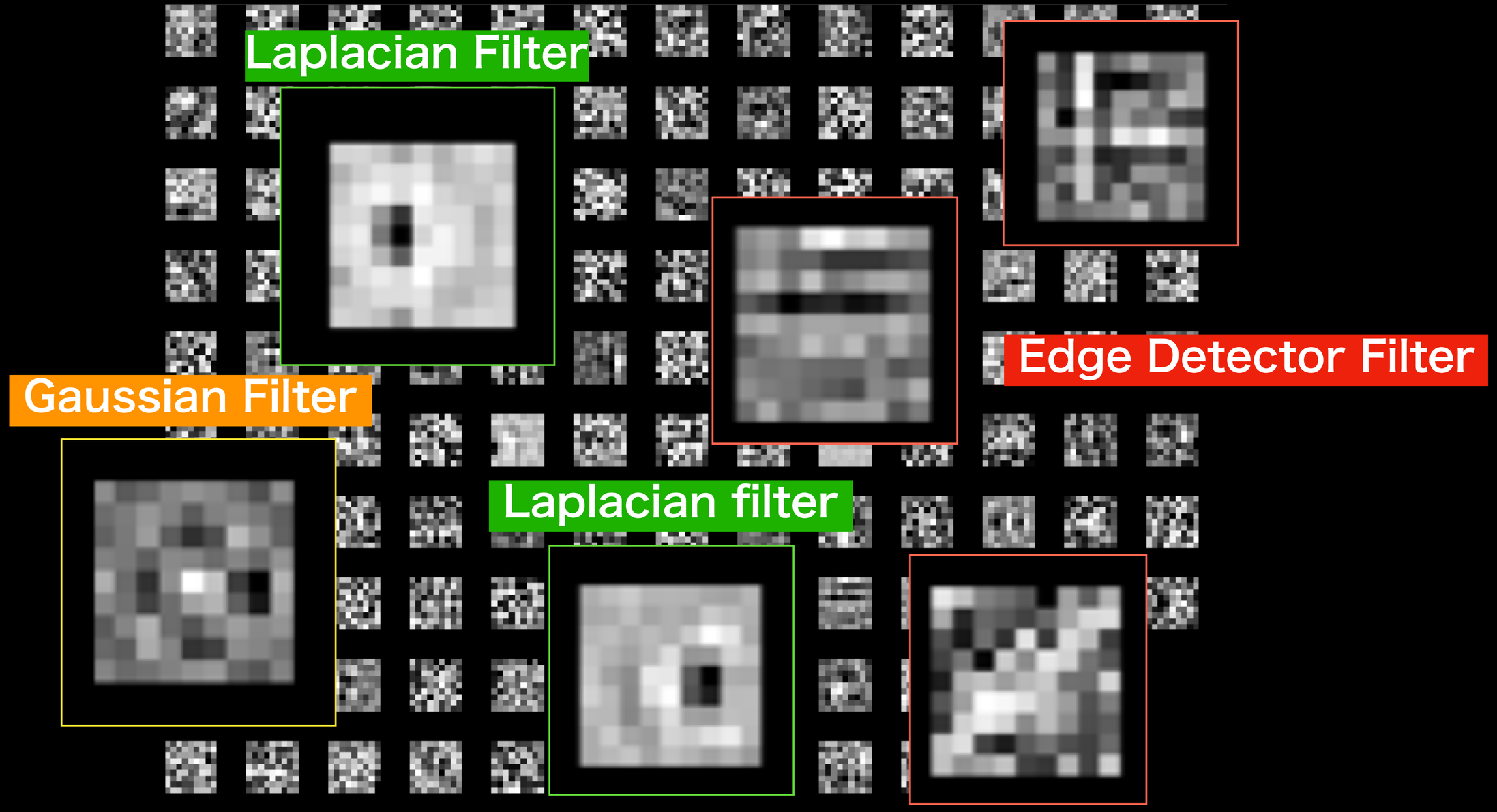
Kernel Weight (after training)



Kernel Weight (after training)

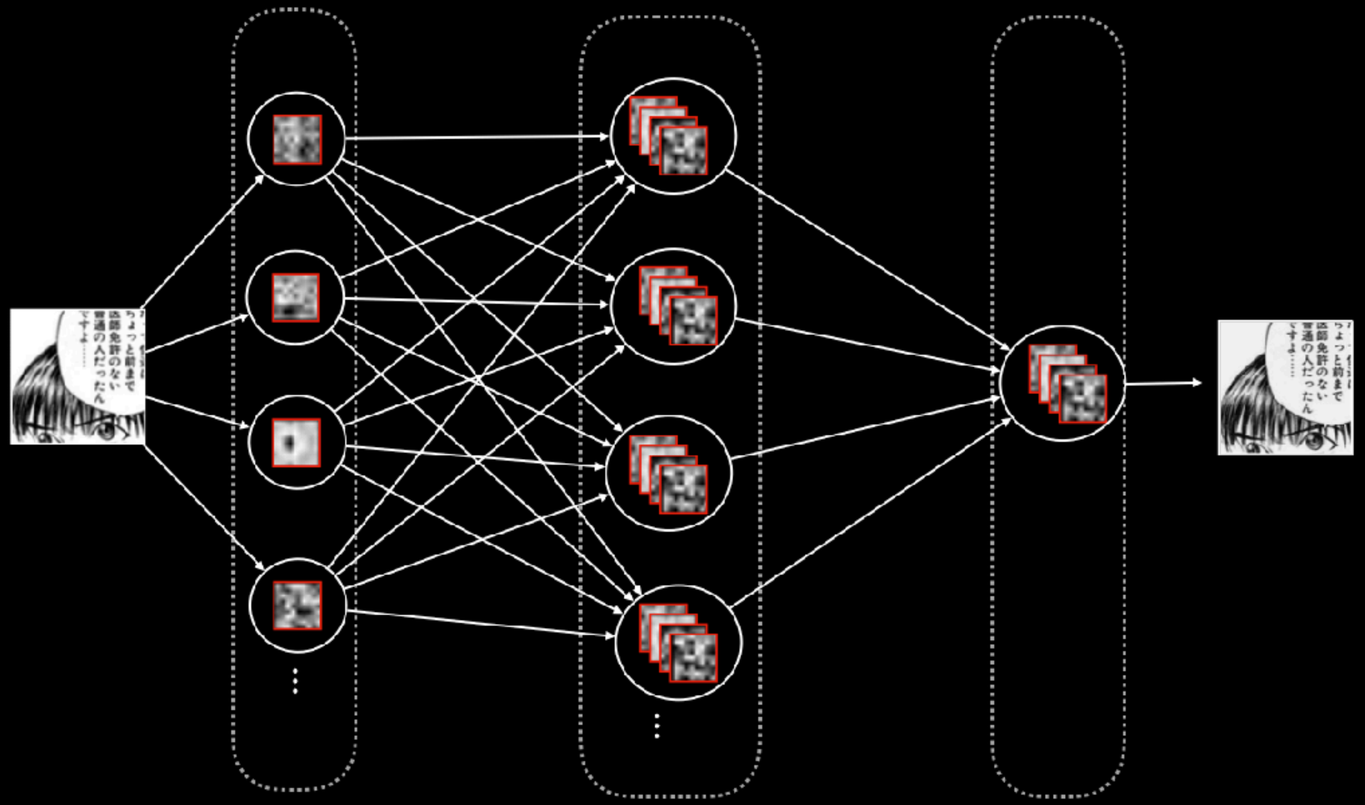


Kernel Weight (after training)



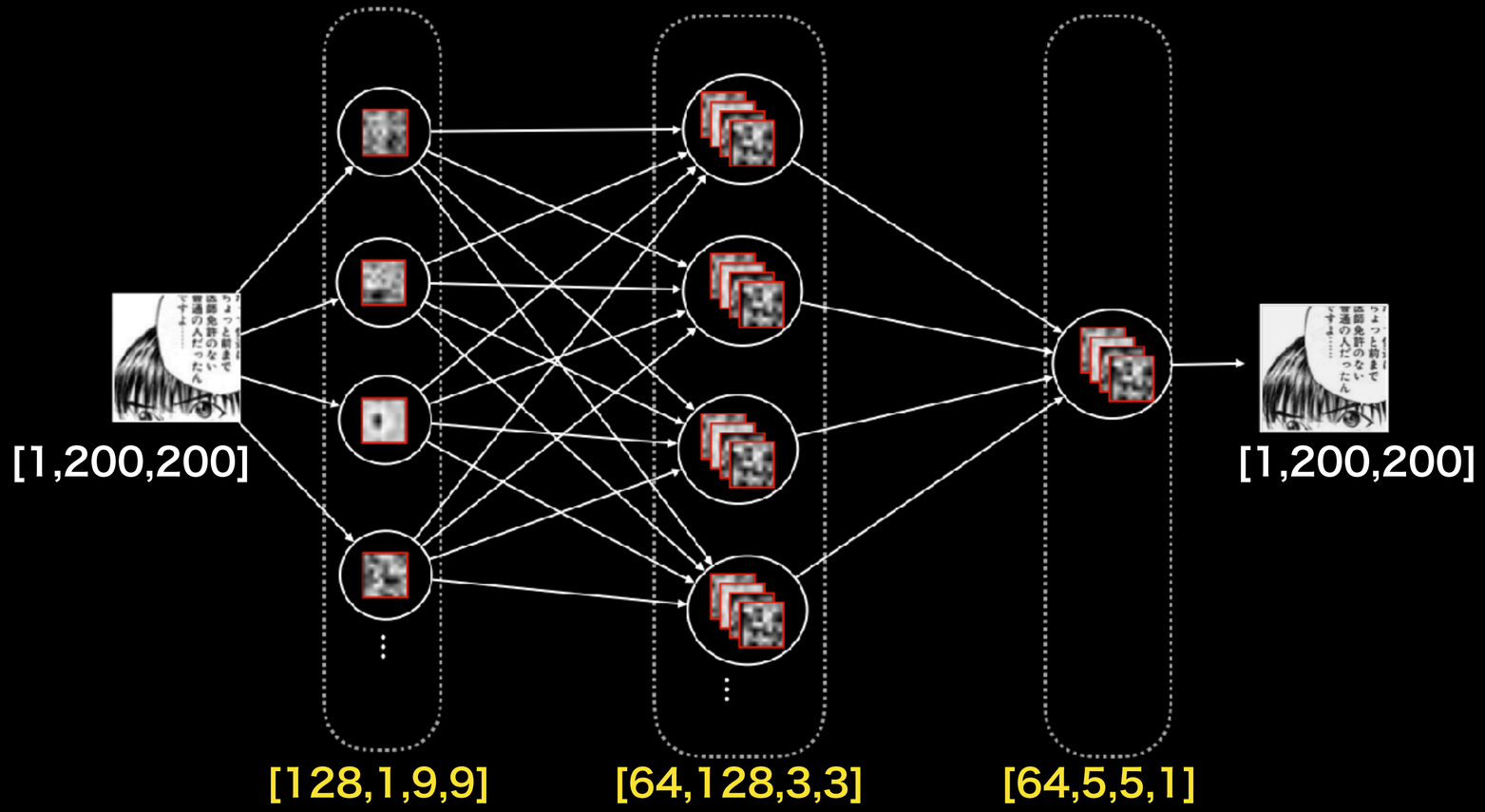
Implementation SRCNN

SRCNN Model Structure



ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

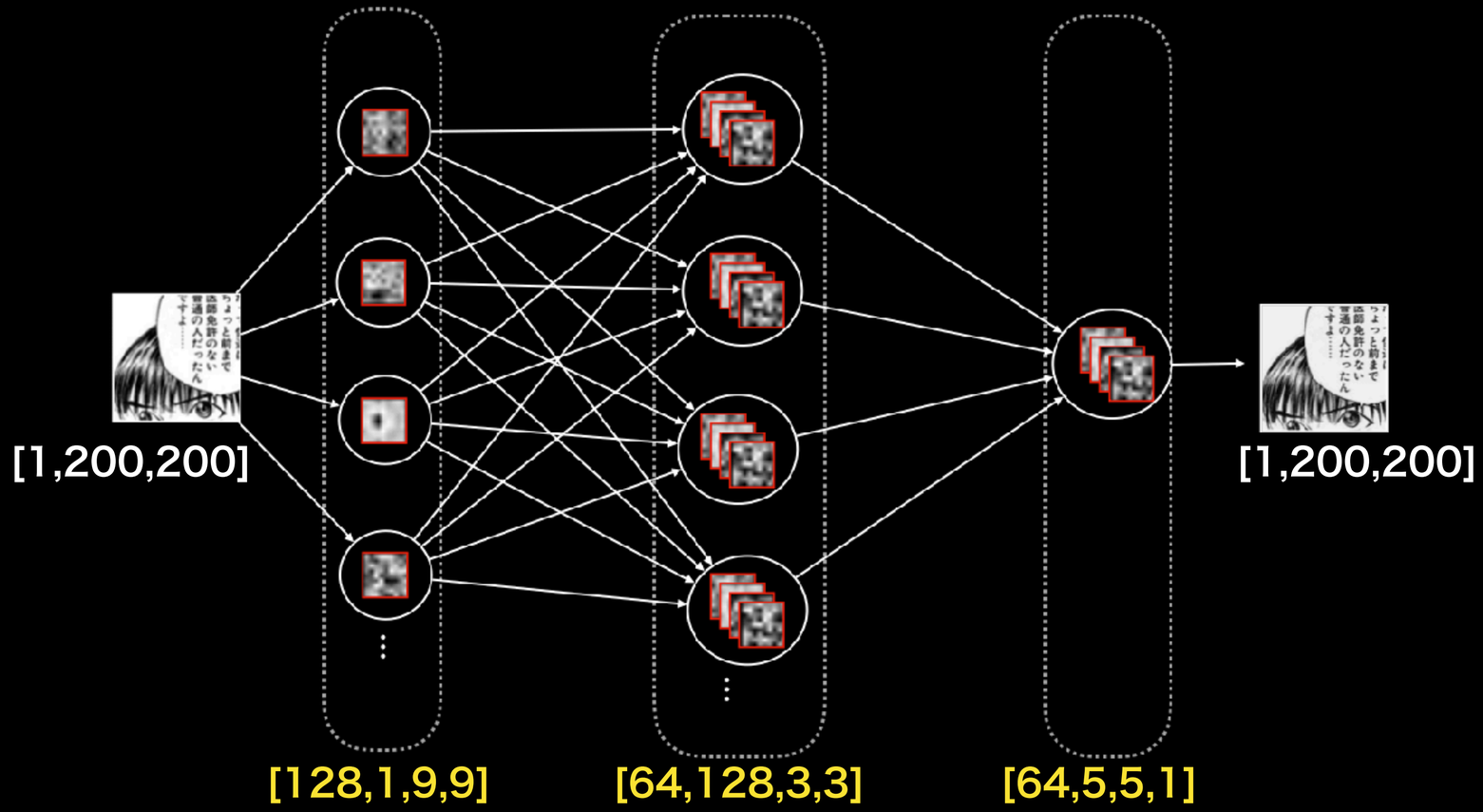
SRCNN Model Structure



ブラックジャックによろしく 佐藤秀峰 Give My Regards to Black Jack SHUHO SATO

SRCNN Model Structure

85,000 parameters

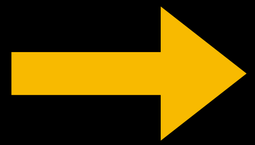


Training code

```
ch = 1
SRCNN = Sequential(input_shape=(200, 200, ch))
SRCNN.add(Conv2D(128, 9, 9, activation='relu'))
SRCNN.add(Conv2D(64, 3, 3, activation='relu'))
SRCNN.add(Conv2D(ch, 5, 5, activation='linear'))
adam = Adam(lr=0.0003)
SRCNN.compile(optimizer=adam, loss='mean_squared_error')
SRCNN.fit(..)
SRCNN.save(model_dir, 'model.h5')
```

Convert to CoreML Model

```
from coremltools.converters.keras import convert
model = convert('model.h5', …)
model.save('SRCNN.mlmodel')
```



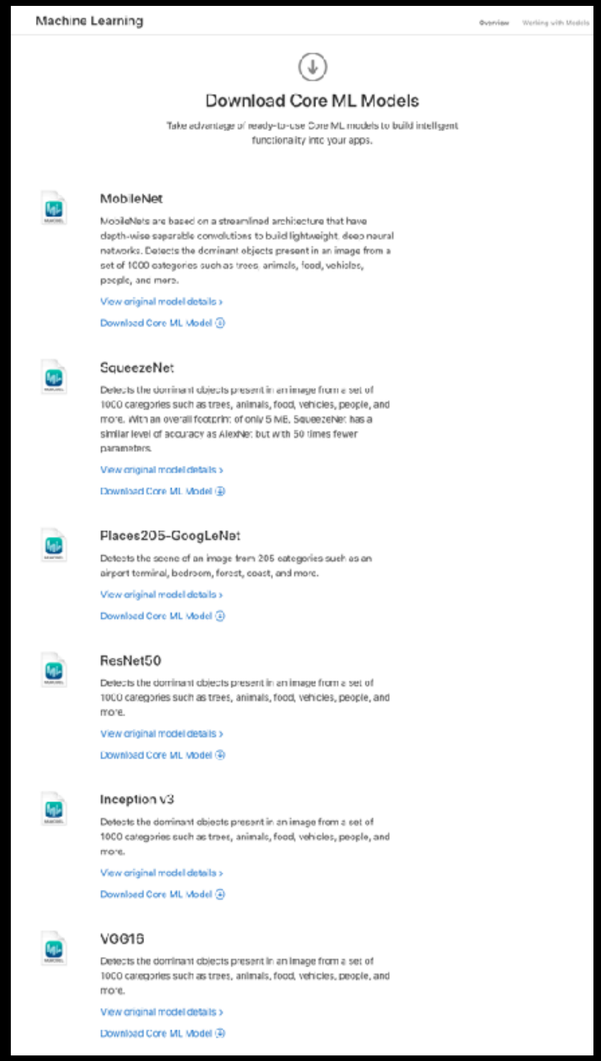
SRCNN.mlmodel

CoreML Model file size

440 KB

SRCNN.mlmodel

<



MobileNet
17 MB

SqueezeNet
5 MB

GoogLeNetPlaces
25 MB

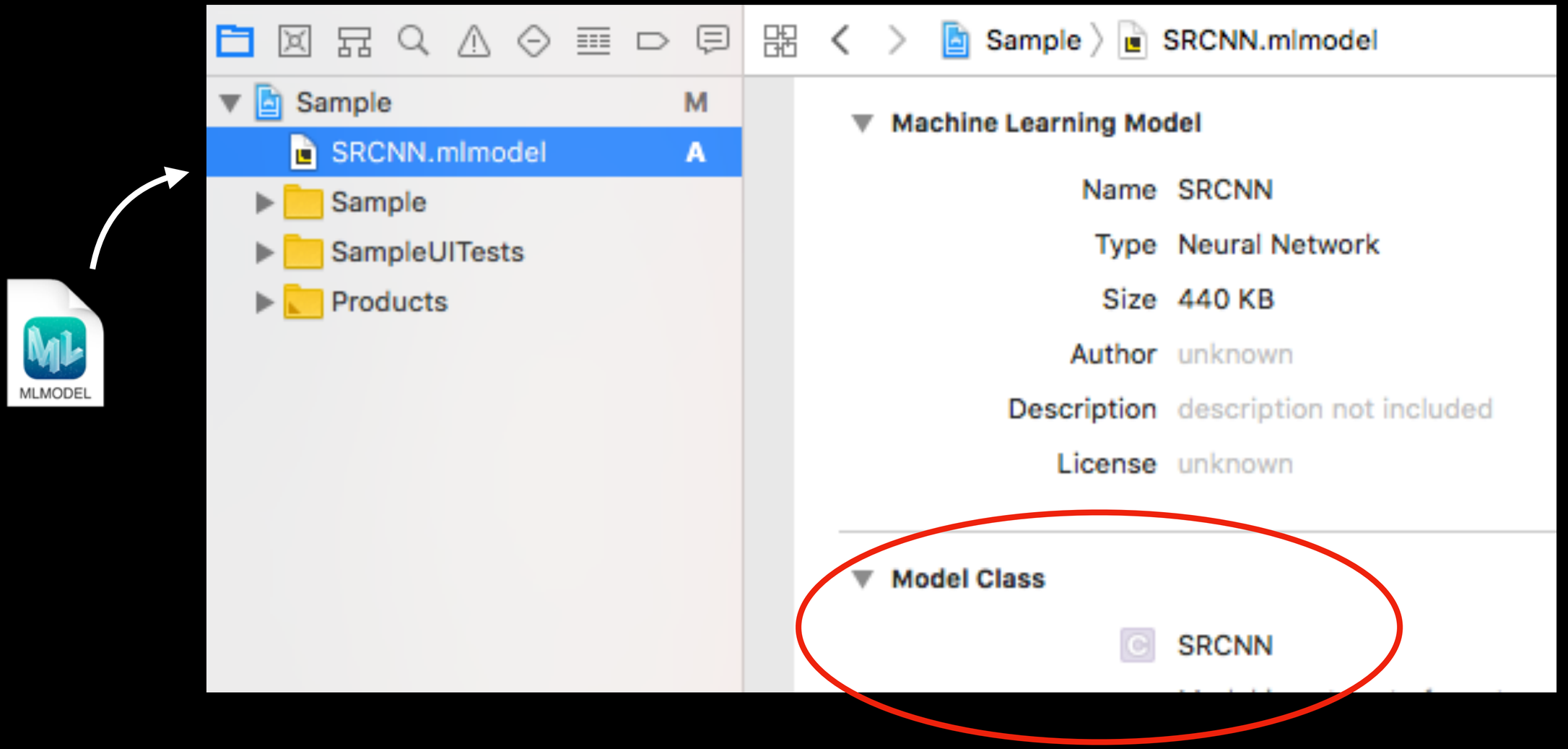
Resnet50
102 MB

InceptionV3
94 MB

VGG16
535 MB

https://developer.apple.com/machine-learning/build-run-models/

Embed CoreML Model into App



Embed CoreML Model into App

```
let model = SRCNN()
let highImage = try! model.predict(image: lowImage)
```

ML Model Accuracy Improvement

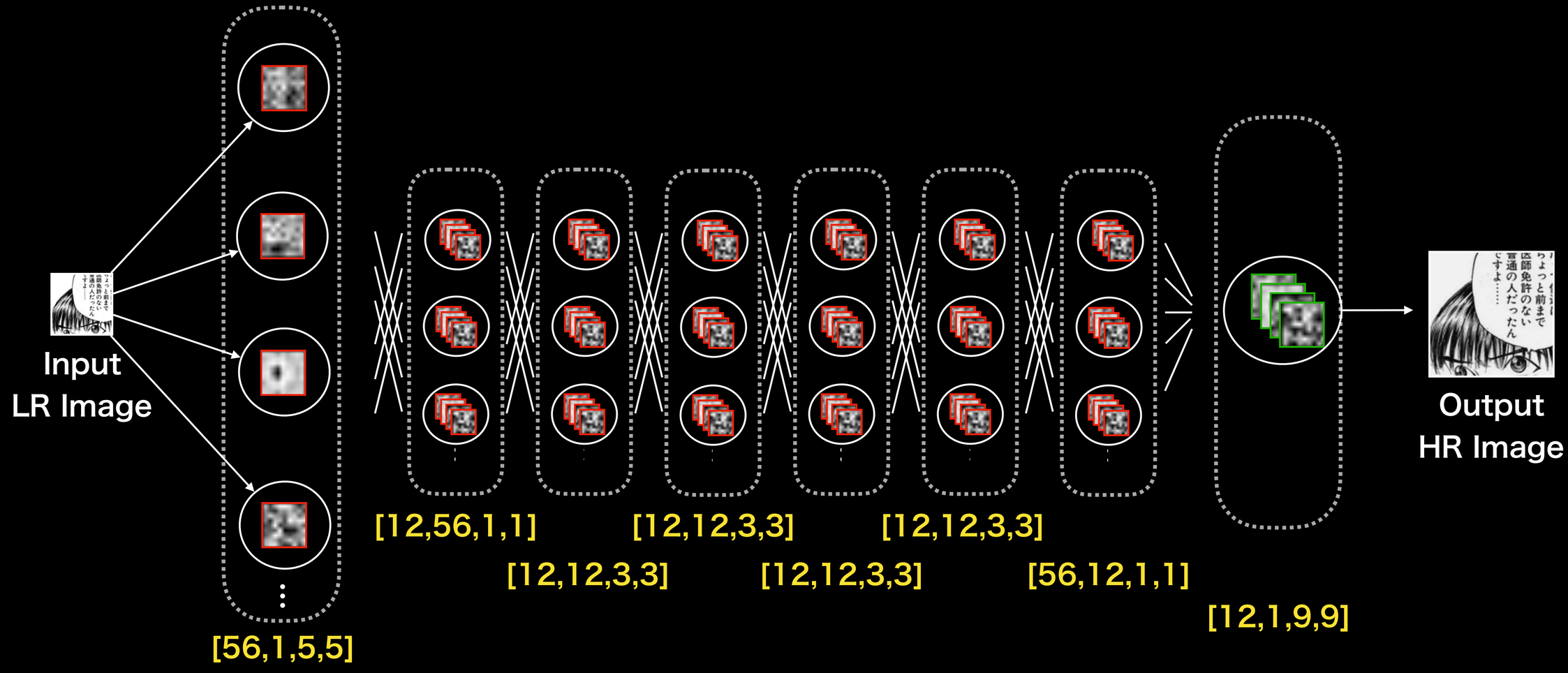
Accelerating the Super-Resolution Convolutional Neural Network

Chao Dong et al. 2016

Fast Super Resolution by CNN (FSRCNN)

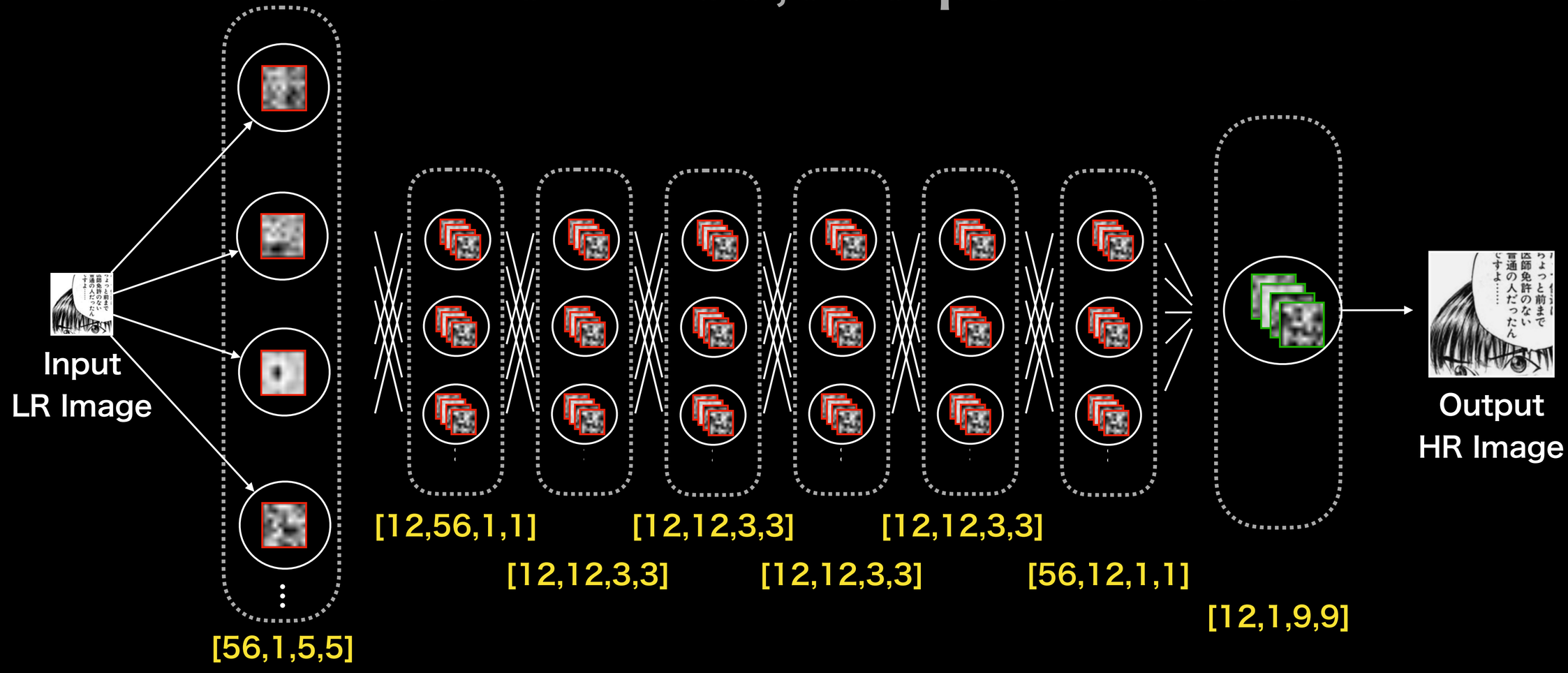
Demo

FSRCNN parameters



FSRCNN parameters

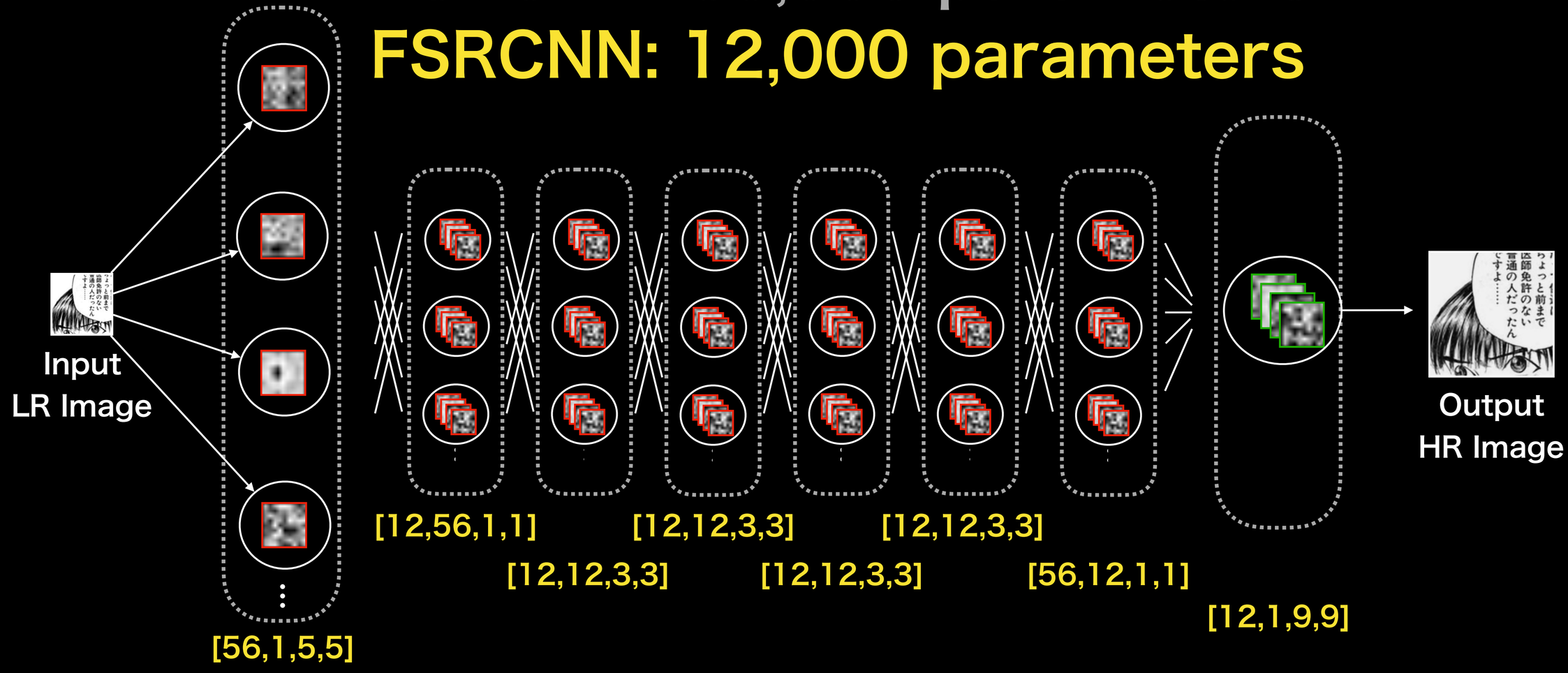
SRCNN: 85,000 parameters



FSRCNN parameters

SRCNN: 85,000 parameters

FSRCNN: 12,000 parameters



Performance

	Bicubic	SRCNN	FSRCNN
Prediction Time/page	0.1sec	1.6sec	1.1sec
Training Time	-	6 hours	1 hours
MLModel Size	-	440 KB	99 KB
PSNR	31.0 db	41.0 db	41.0 db

Recent great papers

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

Christian Ledig et al. 2016

Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network

Wenzhe Shi et al. 2016

Accurate Image Super-Resolution Using Very Deep Convolutional Networks

Jiwon Kim et al. 2016

Pixel Recursive Super Resolution

Ryan Dahl et al. 2017

This is useful for any type of image.
Not only manga.

This is useful for any type of image.
Not only manga.

Give it a try!

https://github.com/kenmaz/SuperResolutionKit

Open Source

https://github.com/kenmaz/SuperResolutionKit

Easy to use

```
pod “SuperResolutionKit”
```

```
let imageView: UIImageView = …
let image: UIImage = …

imageView.setSRImage(image) //Super Resolution⭐
```

SuperResolutionKit

SuperResolutionKit

- UIImageView extension

SuperResolutionKit

- UIImageView extension
- SRConverter (UIImage to UIImage)

SuperResolutionKit

- UIImageView extension
- SRConverter (UIImage to UIImage)
- Include pre-trained model (FSRCNN, SRCNN)

SuperResolutionKit

- UIImageView extension
- SRConverter (UIImage to UIImage)
- Include pre-trained model (FSRCNN, SRCNN)
- Include Python script to train your own model

Limitation

Limitation

- Pre trained model performance

Limitation

- Pre trained model performance
- Quality depends on a types of image

How to leverage ML in Mobile App development

Motivation for ML in Mobile Apps

Motivation for ML in Mobile Apps

Why not just run all this on the server side? 

Motivation for ML in Mobile Apps

Why not just run all this on the server side? 

- Quick response

Motivation for ML in Mobile Apps

Why not just run all this on the server side? 

- Quick response
- Offline support

Motivation for ML in Mobile Apps

Why not just run all this on the server side? 

- Quick response
- Offline support
- Reducing server and network cost

Motivation for ML in Mobile Apps

Why not just run all this on the server side? 

- Quick response
- Offline support
- Reducing server and network cost
- Security

Choose the best way

Technology	Goals	Level
Vision framework	Face / Text Detection, ..	Easy
CreateML	Custom Image / Text Classification, ..	Easy
Turi Create	Custom Object Detection, Style Transfer, ..	Mid
CoreML **(Your own model)**	Others, Cutting edge methods	Hard

Should I learn Python?

Should I learn Python?

- Yes (for now)

Should I learn Python?

- Yes (for now)
  - Python, NumPy, Keras, TensorFlow…

Should I learn Python?

- Yes (for now)
  - Python, NumPy, Keras, TensorFlow…
- Python alternative

Should I learn Python?

- Yes (for now)
  - Python, NumPy, Keras, TensorFlow…
- Python alternative
  - Use Python APIs with Dynamic Member Lookup


```
from keras.models import Sequential
from keras.layers import Conv2D
from keras.optimizers import Adam

ch = 1
SRCNN = Sequential(input_shape=(200, 200, ch))
SRCNN.add(Conv2D(128, 9, 9, activation='relu'))
SRCNN.add(Conv2D(64, 3, 3, activation='relu'))
SRCNN.add(Conv2D(ch, 5, 5, activation='linear'))
adam = Adam(lr=0.0003)
SRCNN.compile(optimizer=adam, loss='mean_squared_error')
```

+ Dynamic Member Lookup

```
let Sequential = Python.import("keras.models.Sequential")
let Conv2D = Python.import("keras.layers.Conv2D")
let Adam = Python.import("keras.optimizers.Adam")

let ch = 1
let SRCNN = Sequential(input_shape: (200, 200, ch))
SRCNN.add(Conv2D(128, (9, 9), activation: 'relu'))
SRCNN.add(Conv2D(64, (3, 3), activation: 'relu'))
SRCNN.add(Conv2D(ch, (5, 5), activation: 'linear'))
let adam = Adam(lr: 0.0003)
SRCNN.compile(optimizer: adam, loss: 'mean_squared_error')
```

Should I learn Python?

- Yes (for now)
  - Python, NumPy, Keras, TensorFlow…
- Python alternative
  - Use Python APIs with Dynamic Member Lookup

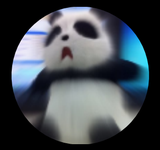
Should I learn Python?

- Yes (for now)
  - Python, NumPy, Keras, TensorFlow…
- Python alternative
  - Use Python APIs with Dynamic Member Lookup
  - Swift for TensorFlow

Recap

- Image Super Resolution details
- How to implement Image Super Resolution with Python and CoreML
- SuperResolutionKit (Please contribute 😎)

Thank You



Twitter / github @kenmaz

https://github.com/kenmaz/SuperResolutionKit