SAIHST
Samsung Advanced Institute for Health Sciences & Technology, SKKU

SAMSUNG
SAMSUNG MEDICAL CENTER

# Classification and Localization of Lesion from Coronary Angiogram Videos via Attention based Weakly-supervised Learning

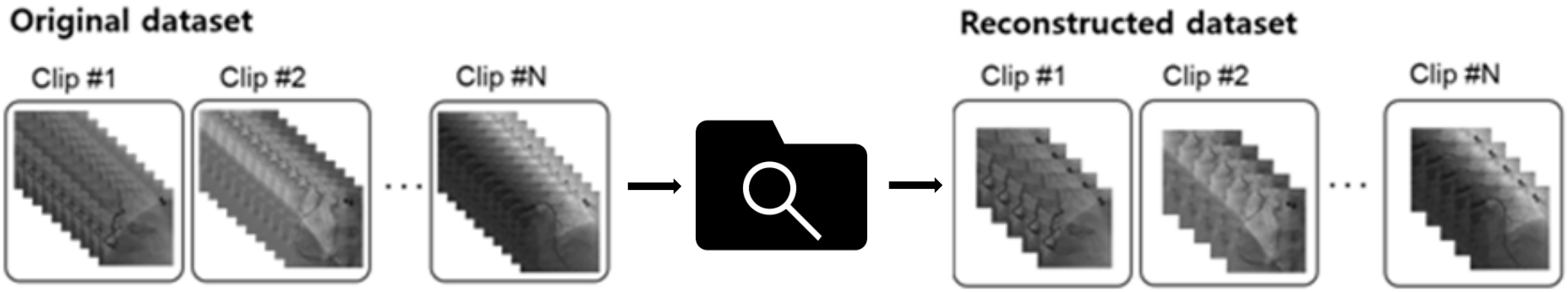
**Jong Hak Moon**

Department of Medical Device Management and Research, SAIHST, Sungkyunkwan University

SUNG KYUN KWAN UNIVERSITY (SKKU)

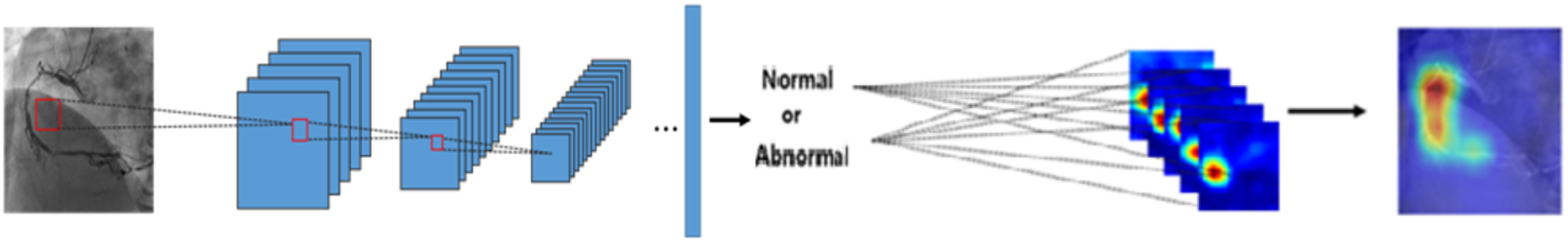
# Overview of our Framework

1st step. Key frame detection using vessel extraction




2nd step. Deep Learning Model Training

3rd step. Qualitative analysis of stenosis

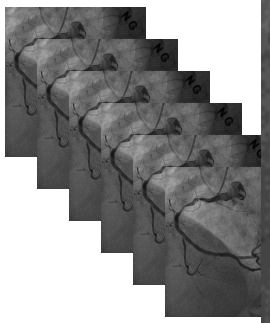


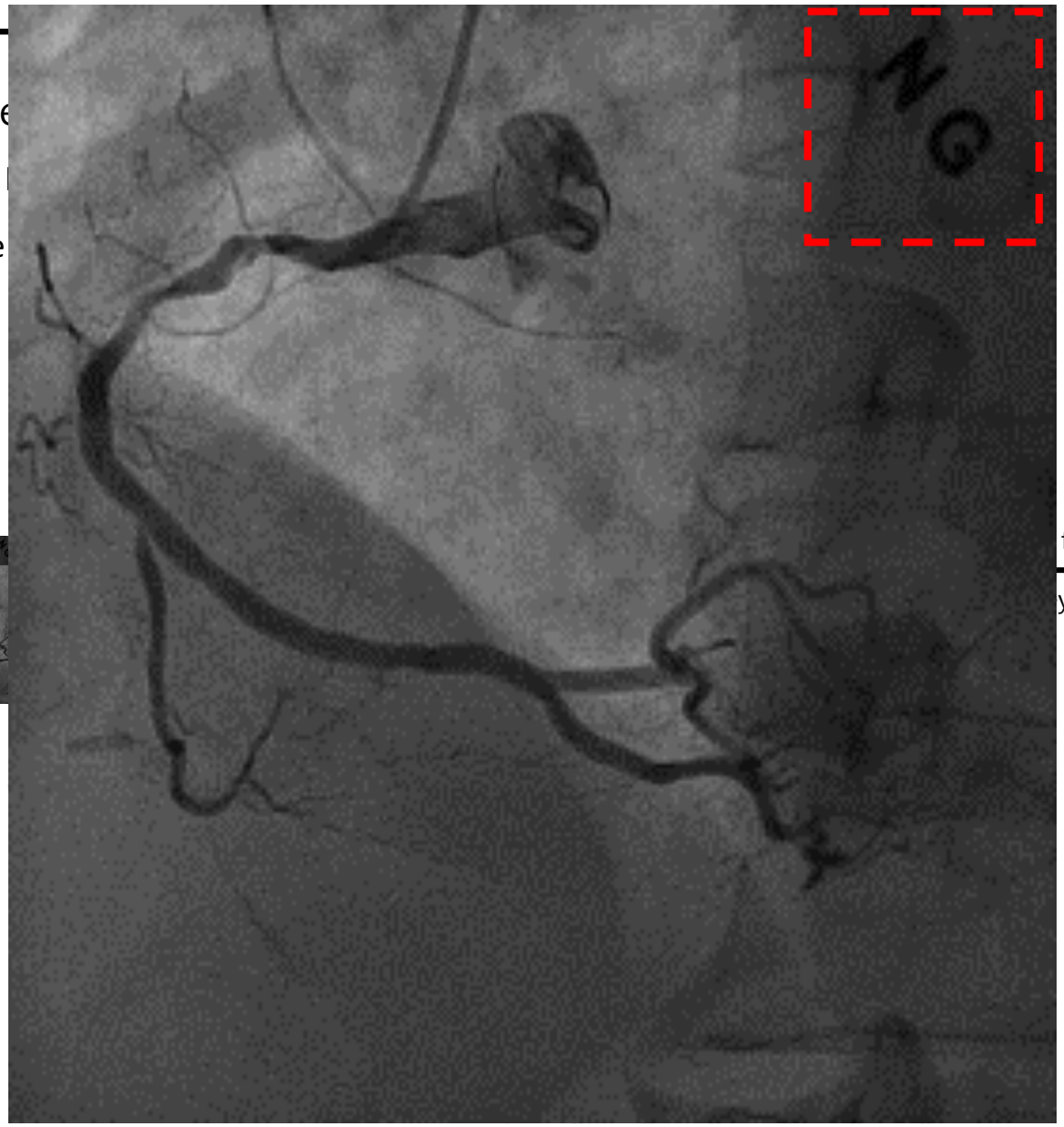
# Text Artifact

## Method –

- ❖ Key Frame
  - → Hessian
  - → Retrieve

Original dataset



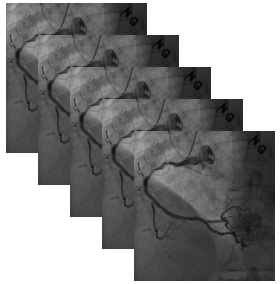



frame

frames

y frame

Reconstructed dataset



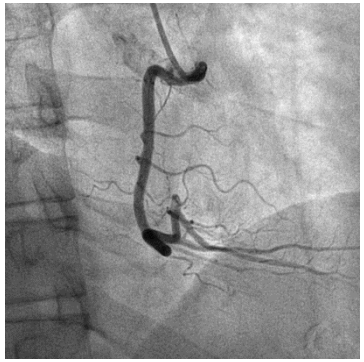
# Method – 2. Data Augmentation

- Data Augmentation (Crop 500x500 , intensity adjustment [-10 to 10])
- Train clips : Validation clips = 361 : 91

  25 times Augmented -> Train clips : Validation clips = 9,025 : 2,275

data size : 512 x 512

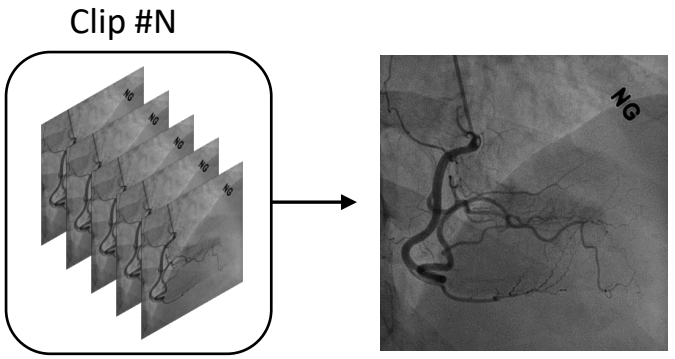
Crop (500x500)
Intensity (-10, 5,0, 5, 10)



data size : 500 x 500

# Training Set up

- **Input Type :** Still image
- **Stratified 5-fold Cross Validation**
- **Key frames in each clip :** 5 frames
- **Epoch** : 10 Epochs
- **1 Self-Attention module**




**< Patient** : 452 patients >

| Data type | | Total<br>Clips & Frames | Normal<br>1.04 | Abnormal<br>1 |
|---|---|---|---|---|
| Original | Number of Clips | 452 Clips | 231 Clips | 221 Clips |
| | Number of Frames | 2,260 Frames | 1,155 Frames | 1,105 Frames |
| Augmented | Number of Clips | 11,300 Clips | 5,775 Clips | 5,525 Clips |
| | Number of Frames | 56,500 Frames | 28,775 Frames | 27,625 Frames |

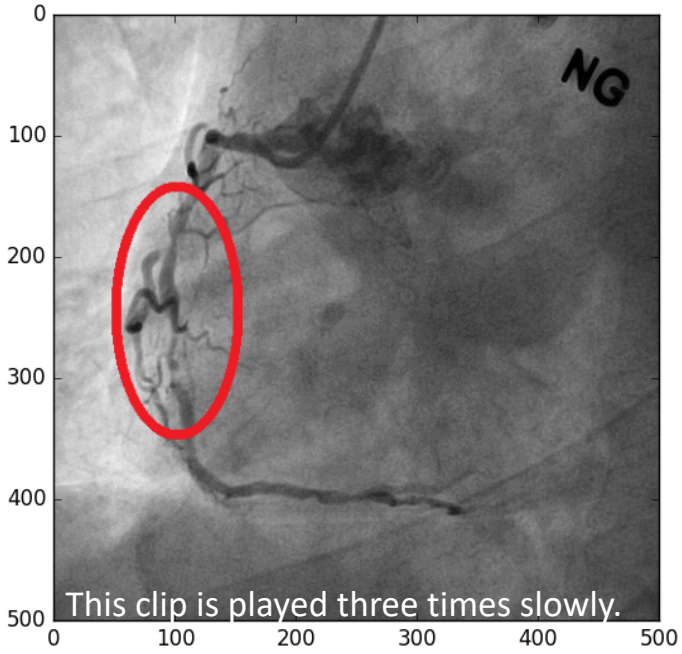
# In-house data validation result

- **Frame-wise Accuracy** : Training depends on number of frames.
- **Clip-wise majority voting test :** Training depends on number of clips.

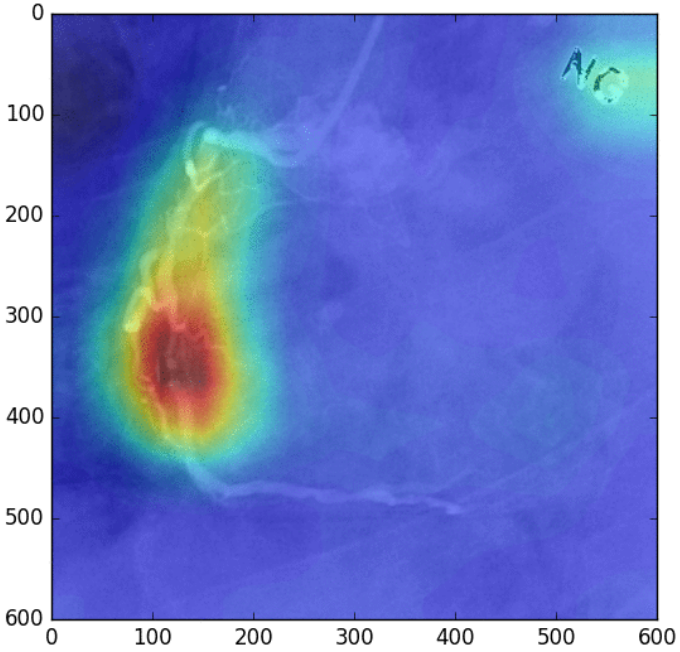
| Training method | AUC (± std.) | Frame-wise Acc (± std.) | Clip-wise Acc (± std.) |
|---|---|---|---|
| Original | 0.958 (± 0.019) | 0.915 (± 0.026) | 0.941 (± 0.025) |
| Original + attention | 0.952 (± 0.016) | 0.920 (± 0.028) | 0.946 (± 0.029) |
| **Augmentation** | **0.968 (± 0.018)** | **0.935 (± 0.026)** | **0.954 (± 0.024)** |
| Aug + attention | 0.964 (± 0.024) | 0.926 (± 0.032) | 0.950 (± 0.031) |

# Qualitative analysis by visualization of feature maps

## Abnormal case (Diffused stenosis)



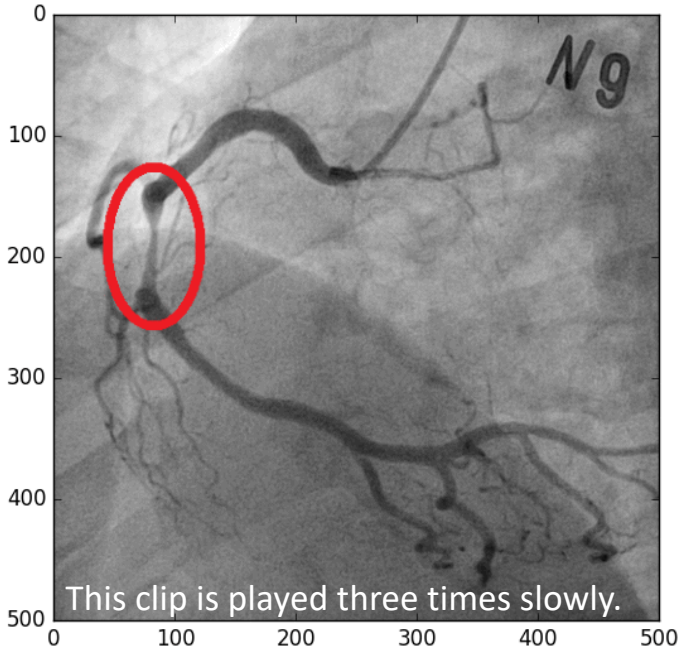

Original Data



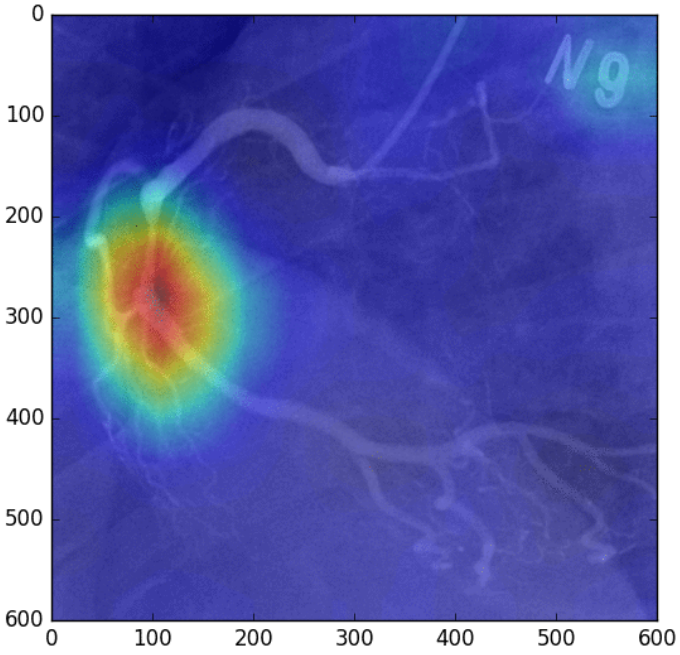

Visualized heat map

# Qualitative analysis by visualization of feature maps

## Abnormal case (Focal Stenosis)



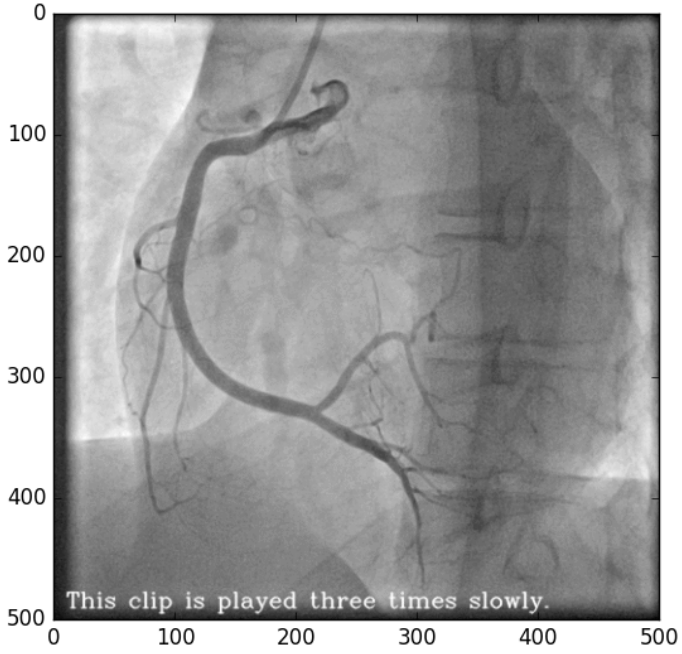

Original Data



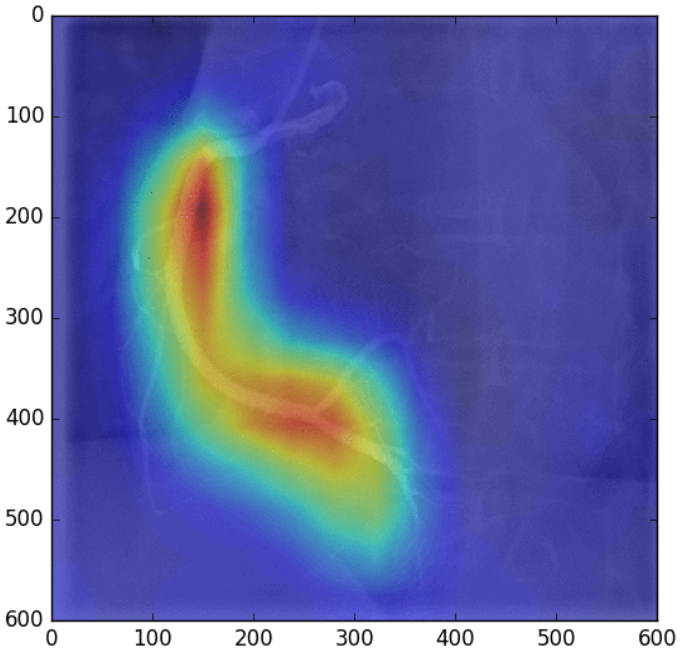

Visualized heat map

# Qualitative analysis by visualization of feature maps

## Normal case




Original Data




Visualized heat map

# CAG AI

## Preparation 1. Module

- TensorFlow >= 1.8.0
- Python >= 3.6.1
  - numpy >= 1.14.3
  - matplotlib >= 2.2.2
  - pillow >= 5.1

- 3분 딥러닝 간단 예제

https://github.com/golbin/TensorFlow-Tutorials

csv

PIL

Cv2

Datetime

Scipy

skimage

Pickle

SimpleITK

Imageio

# CAG AI

## Preparation 2. File structure

Python file steps

1. JH_pickle_generator.py : original data read and make pickle file

2. JH_data_loader_for_kfold.py : pickle file load and split the dataset

   ㄴJH_augment.py : data augmentation

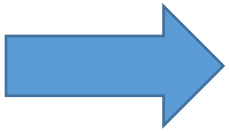
3. JH_main.py : deep learning training and validation based on keras

* Extra file : JH_hyper_parameter_kfold.py -> parameter value tuning

# CAG AI

JH_pickle_generator.py

1. Read Dicom file to numerical value (numpy array)
2. Preprocessing
   - ㄴ Upper right corner occlusion, Contrast Enhancement
   - ㄴ Vesselness filter (Frangi filter)
   - ㄴ Search key frame in mid part and save the index number
3. Pickle file generate
   - ㄴ Load index and reconstruct the dataset with 5 frames in each clip



1 file generate (.dcm to .picke file )

# CAG AI

JH_data_loader_for_kfold.py

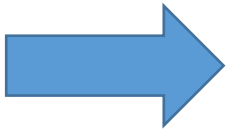
1. Load pickle file
2. Data split and save (k-fold cross validation)

   ㄴTrain, test dataset split and save

   ㄴ Calculate the mean pixel image in train dataset and save

   ㄴ Or applying data augmentation in train data set and calculate mean image

   (crop 500x500, pixel intensity adjustment)



15 (3*5) file generate
(1 pickle file to 5 trainning, validation, and mean image file)

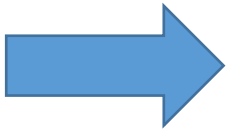
# CAG AI

JH_main.py

1. Transfer learning (Imagenet trained model load)
2. Train Model 구성(deep convolutional netural network load)
3. Pickle file load in each fold train, valid, and subtract mean image file
4. train data = train dataset – mean image / valid = valid – mean image

Train 구성

1. Epoch(augmentation 적용하면 더 많은 횟수 훈련)
2. Train dataset (구성된 딥러닝 모델에 compile 후 예측 모델 생성)



Train and validation accuracy