AI VIETNAM
All-in-One Course
(TA Session)

Introduction to Medical Image Analysis

AI VIET NAM
@aivietnam.edu.vn

Nguyen Thanh Huy

Outline

- Medical Image Analysis in General
- Medical Research Team – AIMI (AI VIETNAM)
- Medical Image Processing
- Deep Learning on Medical Imaging
- Research Project: Breast Ultrasound Cancer Diagnosis
- Research Idea Completion and Research Directions
- Exercises

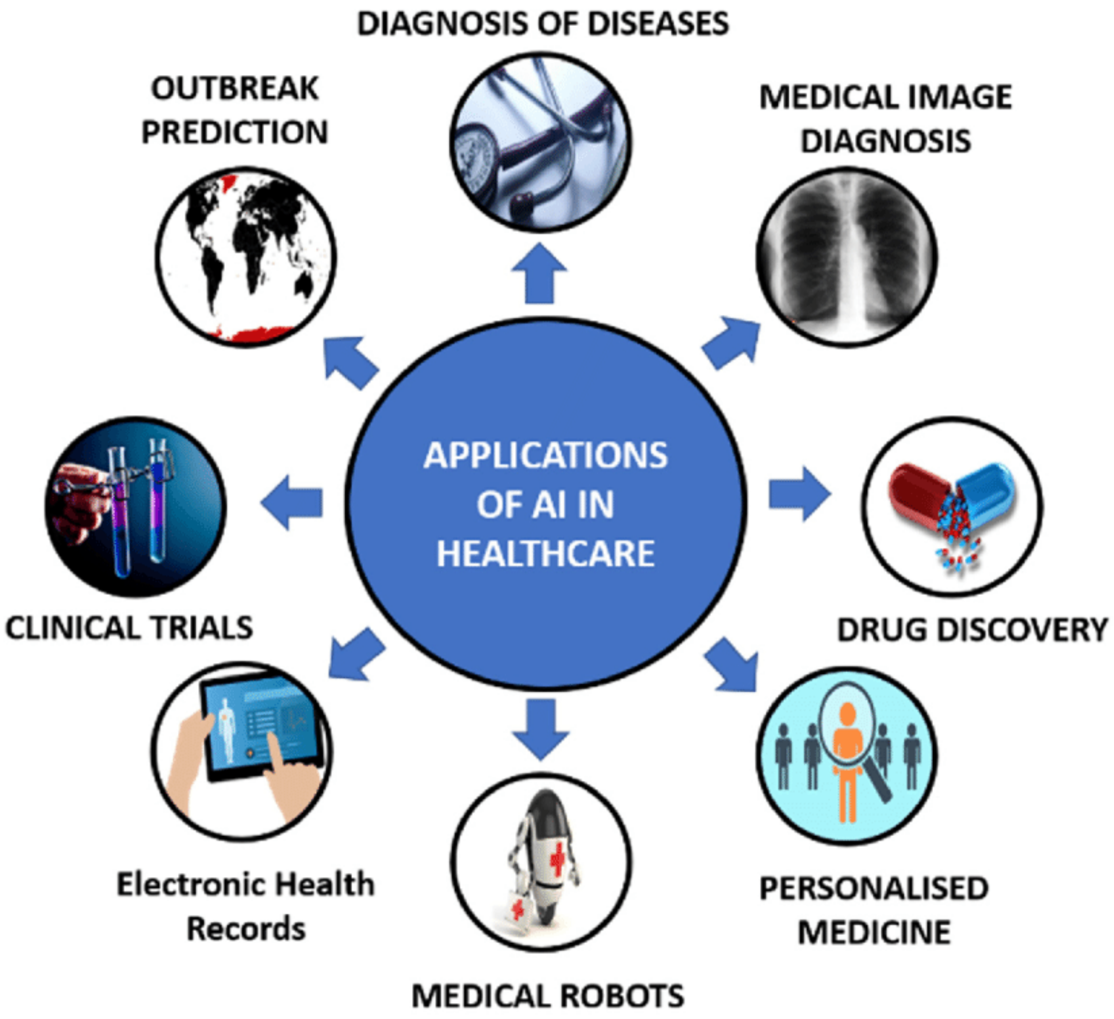
AI VIETNAM
All-in-One Course
(TA Session)

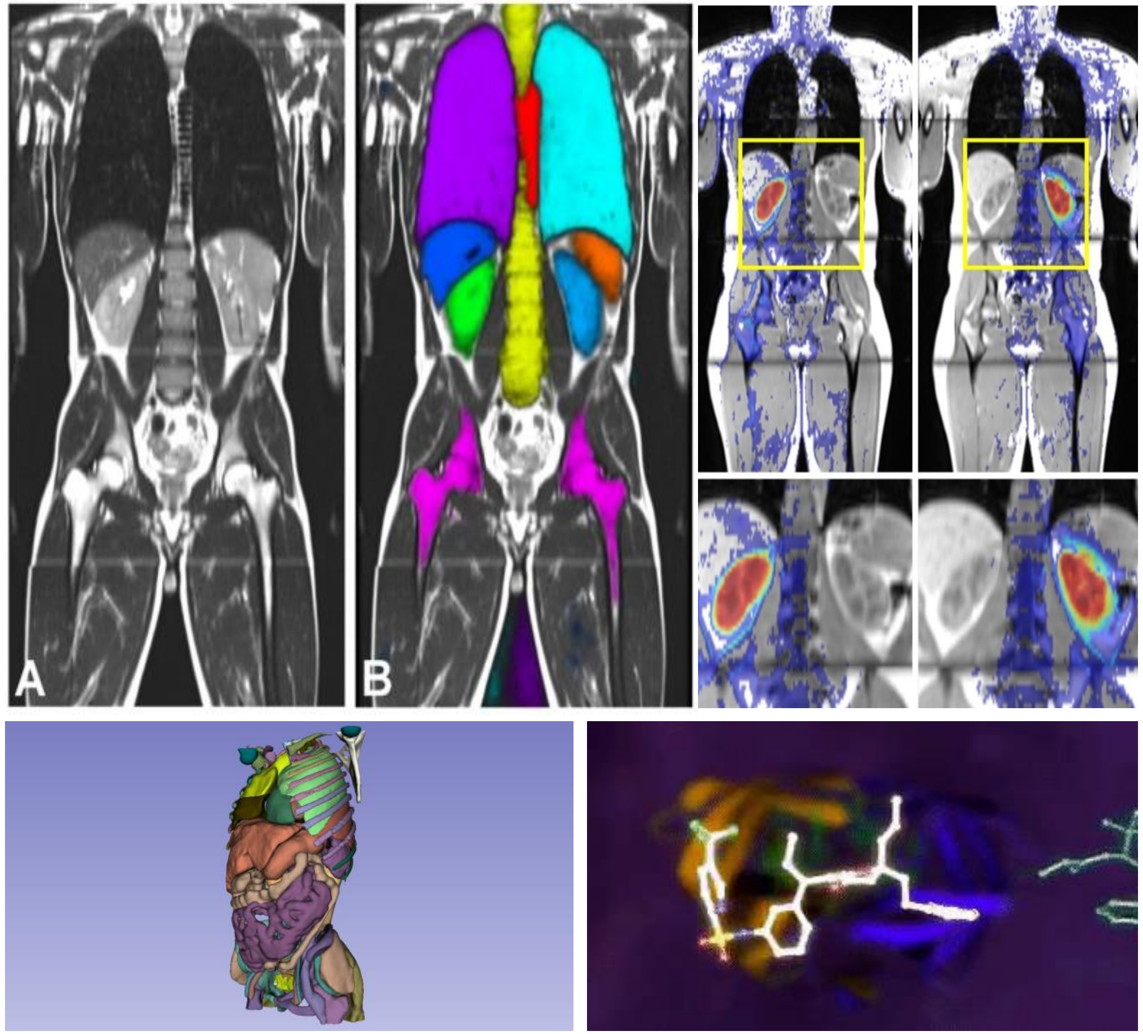
Medical Image Analysis in General

Nguyen Thanh Huy

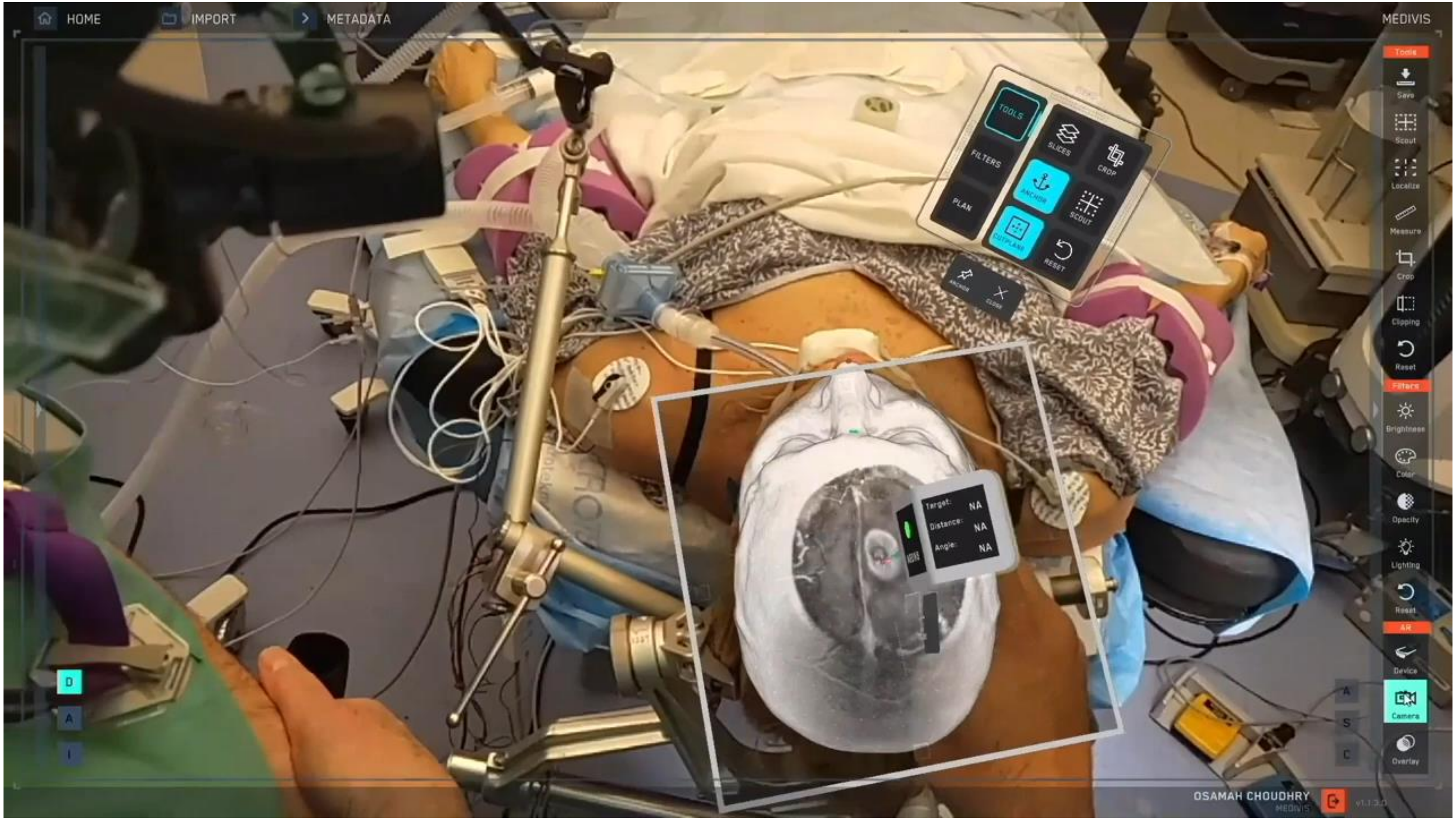
Medical Applications

❖ Overview of Applications



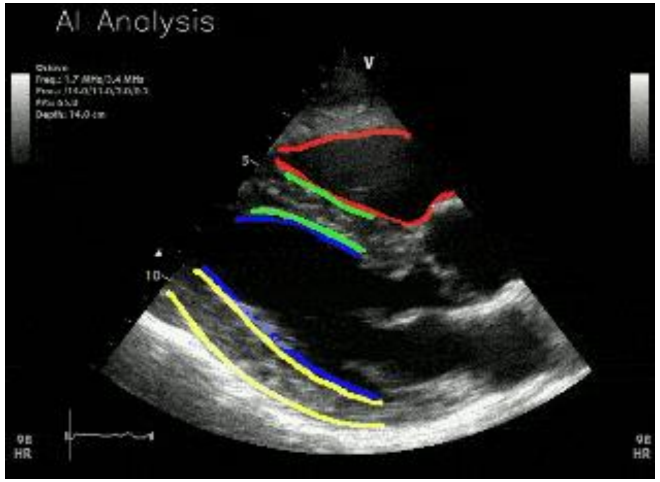


Medical Applications

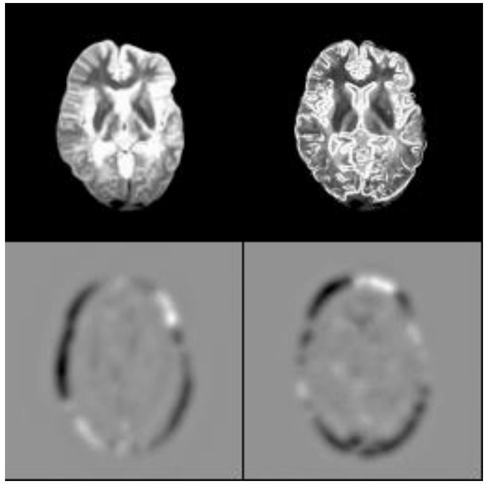


Medical Image Modalities

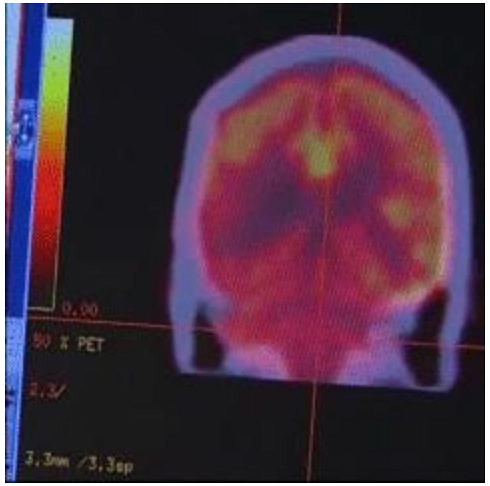
X-Ray



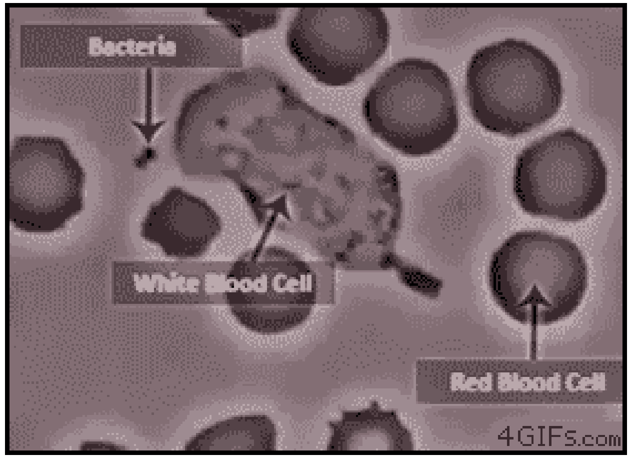
Ultrasound



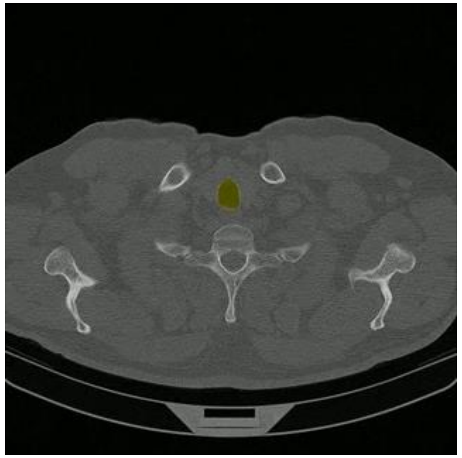
MRI



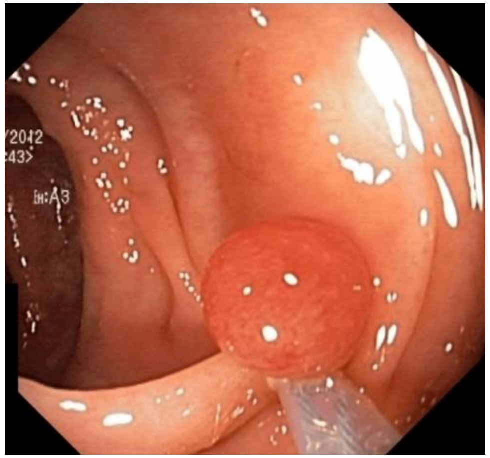
PET



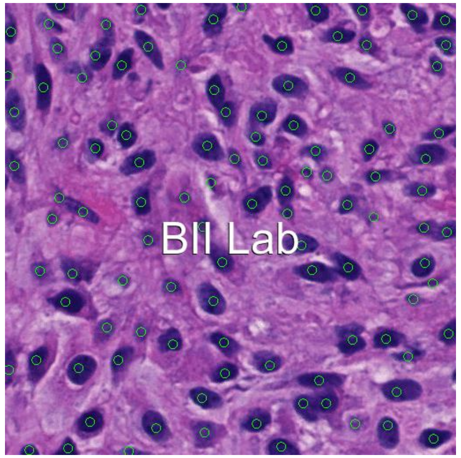
Microscopy



CT



Endoscopy



Histopathology

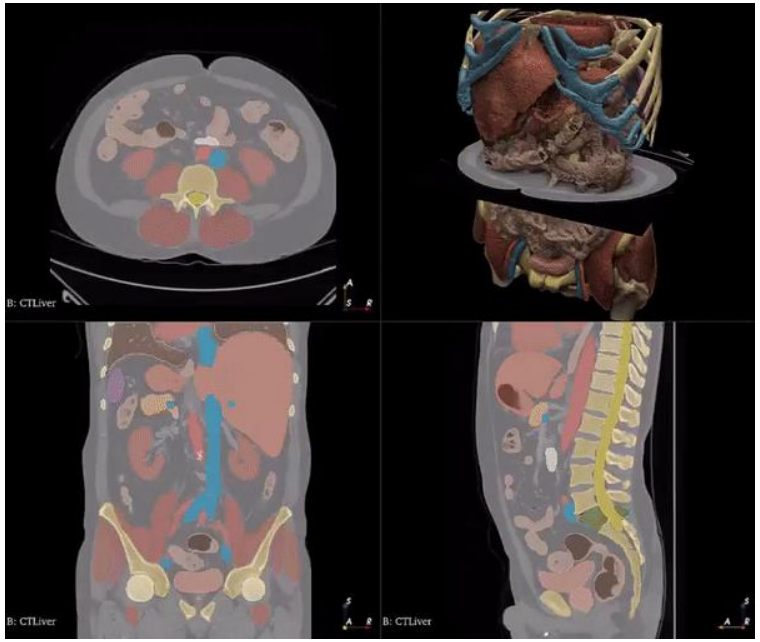
AI Tasks in Medical Imaging

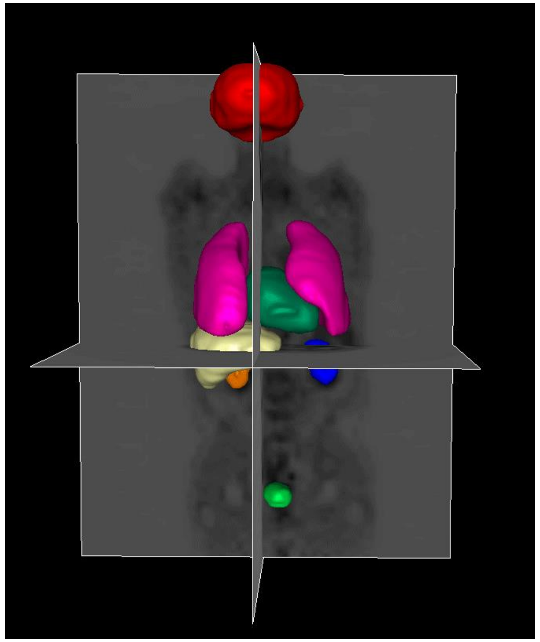
2D Modalities:
- ❖ Classification
- ❖ Detection
- ❖ Segmentation

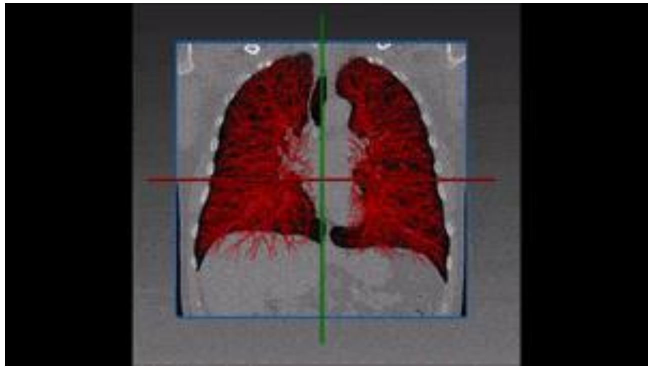
3D Modalities:
- ❖ Registration
- ❖ Domain Conversion

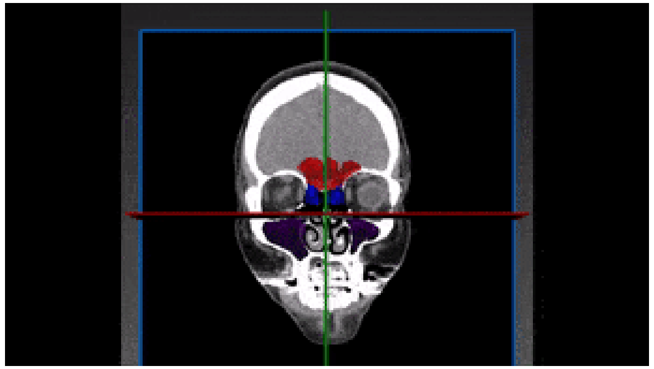
Video-based:
- ❖ Tracking
- ❖ Prognosis (Future Prediction)
- ❖ 3D Reconstruction

Other Tasks:
- ❖ Quality Enhancement
- ❖ Multi-modalities
- ❖ Synthesis









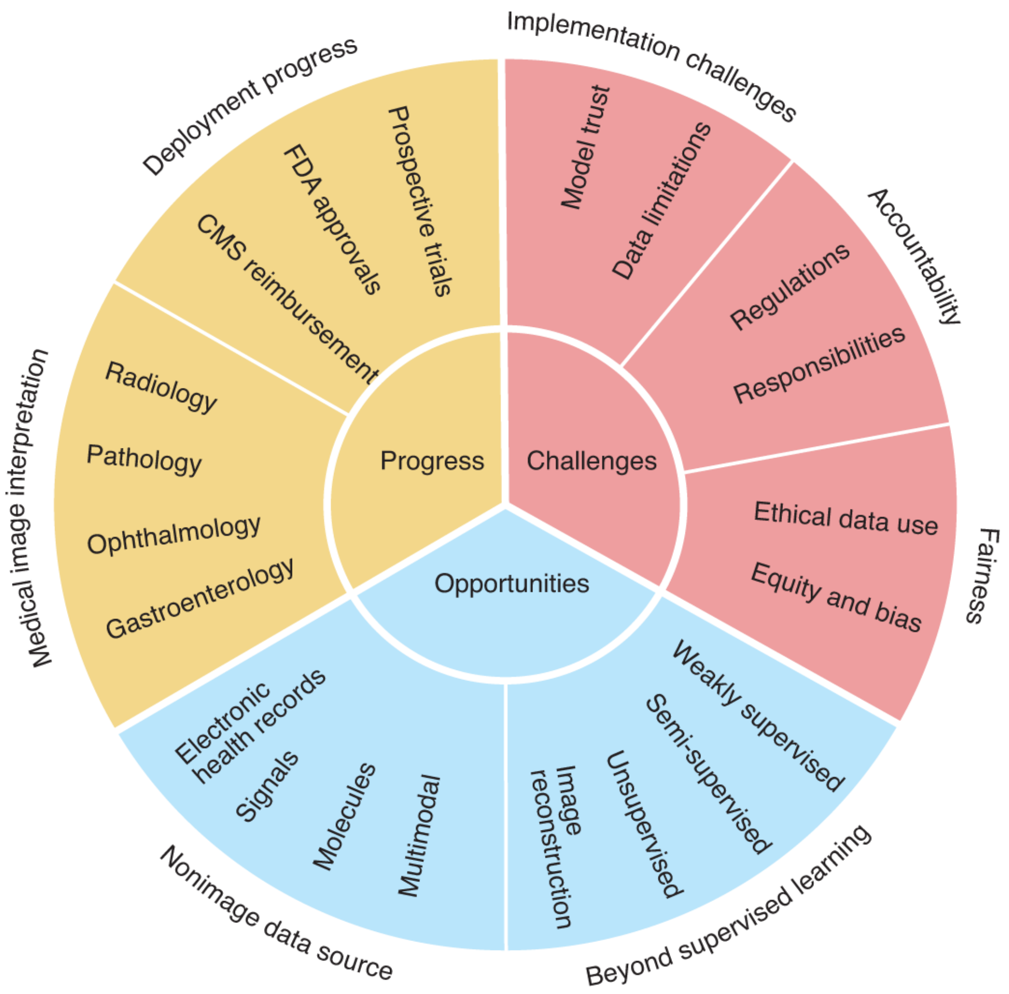
Medical Domain Challenges

Data Limitation:
❖ Time-consuming for labelling, experts required.
❖ Lack of public data (Private data is better).
❖ Data is low quality, missing

Model Reliability:
❖ Model bias should be concerned for workflow.
❖ Protocol for labelling procedure is a must.
❖ Domain gap between multiple subjects.
❖ Conclusion based on multiple data types.

Fairness:
❖ Private data license is required (equity)
❖ Medical bias while labeling.

Other Challenges:
❖ Accountability (Regulations & Responsibilities)



Medical Domain Challenges

Data Limitation:
- ❖ Time-consuming for labelling, experts required.
- ❖ Lack of public data (Private data is better).
- ❖ Data is low quality, missing

Model Reliability:
- ❖ Model bias should be concerned for workflow.
- ❖ Protocol for labelling procedure is a must.
- ❖ Domain gap between multiple subjects.
- ❖ Conclusion based on multiple data types.

Fairness:
- ❖ Private data license is required (equity)
- ❖ Medical bias while labeling.

Other Challenges:
- ❖ Accountability (Regulations & Responsibilities)

Technical Solutions:
- ❖ Active Learning
- ❖ Unsupervised, Semi-supervised
- ❖ Image Restoration, Enhancement
- ❖ Noisy labeling
- ❖ Weakly-Supervised
- ❖ Domain Adaptation, Generalization
- ❖ Multi-modality
- ❖ Federated learning
- ❖ Explainable AI

- ➢ Trustworthy model is a must for clinical implementation
- ➢ MedAI is deployed to assist doctors and radiologists, not replacing them

Medical Research Team – AIMI (AI VIETNAM)

Nguyen Thanh Huy

AI in Medical Imaging (AIMI) – AI VIETNAM

AIMI Research Group Information:

- ❖ Leader: Thanh Huy
- ❖ Foundation: April 2023
- ❖ Achievements:
  - ➢ 8 Accepted conference/journal papers.
  - ➢ 6 Submitted conference/journal papers.
  - ➢ Total: 2 Q1 journals, 1 rank A*, 4 rank A conf
- ❖ Research Networks:
  - ➢ National Cheng Kung University (TW)
  - ➢ Taipei Medical University (TW)
  - ➢ National Taiwan University (TW)
  - ➢ National Central University (TW)
  - ➢ Nanyang Technological University (SG)
  - ➢ National Skin Center (SG)
  - ➢ Saigonmec (VN)
  - ➢ Blood Transfusion and Hematology Center (VN)
  - ➢ Other Domestic Universities (VN)

AIMI Publications

- (ICCV 2023): Nguyen, Thanh-Huy, et al. "Towards Robust Natural-Looking Mammography Lesion Synthesis on Ipsilateral Dual-Views Breast Cancer Analysis." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023.
- (MICCAI 2023): Truong, T. T., Nguyen, H. T., Lam, T. B., Nguyen, D. V., & Nguyen, P. H. (2023, October). Delving into Ipsilateral Mammogram Assessment Under Multi-view Network. In International Workshop on Machine Learning in Medical Imaging (pp. 367-376). Cham: Springer Nature Switzerland.
- (IEEE SSP 2023): H. T. Nguyen, T. B. Lam, Q. T. D. Tran, M. T. Nguyen, D. T. Chung and V. Q. Dinh, "In-context Cross-Density Adaptation on Noisy Mammogram Abnormalities Detection," 2023 IEEE Statistical Signal Processing Workshop (SSP), Hanoi, Vietnam, 2023, pp. 383-387.
- (TAAI 2023): Gia-Van To, Thanh-Huy Nguyen, Manh-The Nguyen. SMOTE-MD: Synthetic Algorithm using Mahalanobis Distance for Casualty Insurance.
- (TAAI 2023): Quan Dinh Dai Tran, Toan Thai Ngoc Truong, Quoc-Vinh Luu, Anh H. Dao, Minh T. Nguyen, and Quang-Vinh Dinh. Delineating COVID-19 Pulmonary Infiltrate Manifestation Leveraging Auxiliary Tasks
- (TAAI 2023): Thinh B. Lam, Hien Q. Kha, Huy T. Nguyen, Dinh-Tan Nguyen, Quan D. Nguyen, Toan T. N. Truong, Manh D. Vu, Nguyen Quoc Khanh Le. Redesigned Dual-Task Learning Framework for Diagnosis Mammography Screening with BI-RADS and Density Classification.
- (ICISN 2024): Quoc-Vinh Luu, Khanh-Duy Le, Thanh-Huy Nguyen, Thanh-Minh Nguyen, Tien-Thinh Nguyen, and Quang-Vinh Dinh. (2024) Semi-supervised Semantic Segmentation using Redesigned Self-Training for White Blood Cells.
- (ICISN 2024): Dat T. Chung, Minh-Anh Dang, Mai-Anh Vu, Minh T. Nguyen, Thanh-Huy Nguyen, and Vinh Q. Dinh. (2024) Beyond Traditional Approaches: Multi-Task Network for Breast Ultrasound Diagnosis.
- (ISBI 2024): Thanh-Huy Nguyen, Thi Kim Ngan Ngo, Mai Anh Vu, Ting-Yuan Tu. Blurry Consistency Segmentation Framework with In-focus Spatial Stacking on 3D Breast Cancer Cell.
- (ISBI 2024): , Hien Q. Kha, Dinh T. Nguyen, Thinh B. Lam, Thanh-Huy Nguyen, Cao T. Tran, Manh D. Vu, Lan T. Ho-Pham, Liem Pham, Nguyen Quoc Khanh Le. (2024) M2Net: Two-stage Multi-label Breast Cancer Detection Networks.
- (TNU Journal): Thuy Phuong Nhu Le, Thanh-Huy Nguyen. Using convolutional neural network (CNN) for COVID-19 chest X-ray diagnosis.
- (Q1 Journal): Ba Hung Ngo, Ba Thinh Lam, Thanh-Huy Nguyen, Quang Vinh Dinh, Tae Jong Choi. Dual Dynamic Consistency Regularization for Semi-supervised Domain Adaptation.
- (Q1 Journal): Toan T. Nguyen, Huy T. Nguyen, Huy Q. Ung, Hieu T. Ung and Binh T. Nguyen. Deep-Wide Learning Assistance for Insect Pest Classification.

Red: AIO 2021 Green: AIO 2022 Blue: AIO 2023

Research Projects

AIMI Research Projects:

- ❖ 3D Microscopy Breast Cancer Cell Tracking (Affiliated with NCKU)
- ❖ Drug Response Prediction on Lung Cancer Cell Lines (Affiliated with TMU)
- ❖ MRI-based 3D Liver and Cancer Segmentation. (Affiliated with TMU)
- ❖ Mammography Abnormalities Detection (Affiliated with Saigonmec x TMU)
- ❖ Skin Lesions Classification and Segmentation (Affiliated with NTU-SG x NSC)
- ❖ Cardiac Signal Diseases Prediction (Affiliated with NTU-TW x Harvard x Yale)
- ❖ White Blood Cell Segmentation and Subtypes Recognition (Affiliated with BTH)
- ❖ Semi-supervised Biomedical Segmentation (AIMI)

NATIONAL CHENG KUNG UNIVERSITY 1931

TAIPEI MEDICAL UNIVERSITY 1960

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

HARVARD T.H. CHAN SCHOOL OF PUBLIC HEALTH

Yale University School of Medicine

SAIGONMEC

BTH BỆNH VIỆN TRUYỀN MÁU HUYẾT HỌC

Looking for AIO 2023 Research Members

AI VIET NAM
@aivietnam.edu.vn

Requirement:

❖ Self-motivated and independent research members.

❖ Active and curious research members.

❖ Pursuing the long-term goal towards AI in medical field.

❖ Good technical skill (implement and debug) is a big plus

❖ No need prior research experience.

Benefit:

❖ Be trained to equip all core research mindsets and skills.

❖ Boosting research profile and achievements (especially for higher education)

❖ GPU supported (more than one AIVIETNAM 24GB, one AIMI 24GB, Project extra*)

❖ Publication fee partially or fully supported (depends on impact of venue)

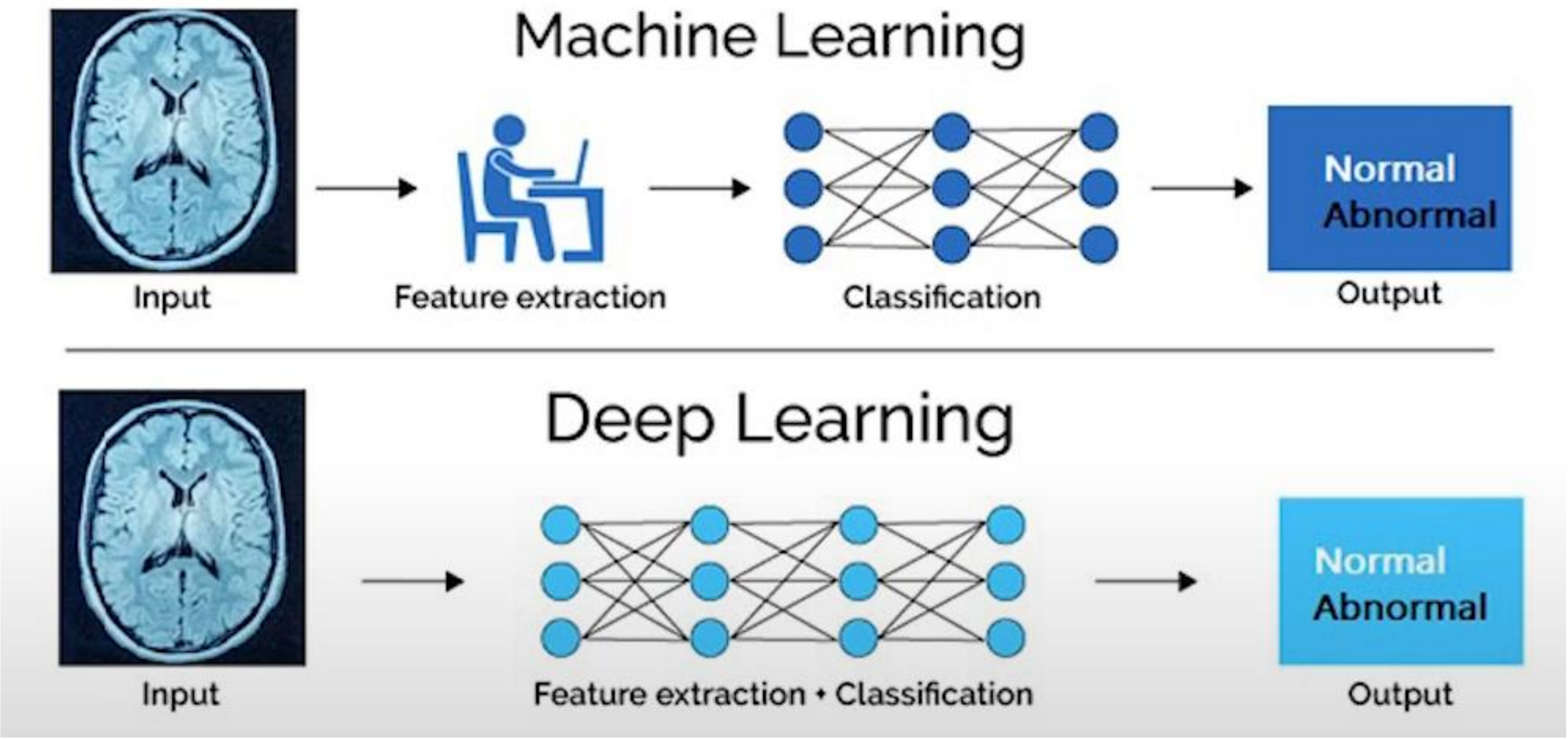
❖ Letter of Recommendation; Reference for MSc, PhD position if needed.

Medical Image Processing

AI VIET NAM
@aivietnam.edu.vn

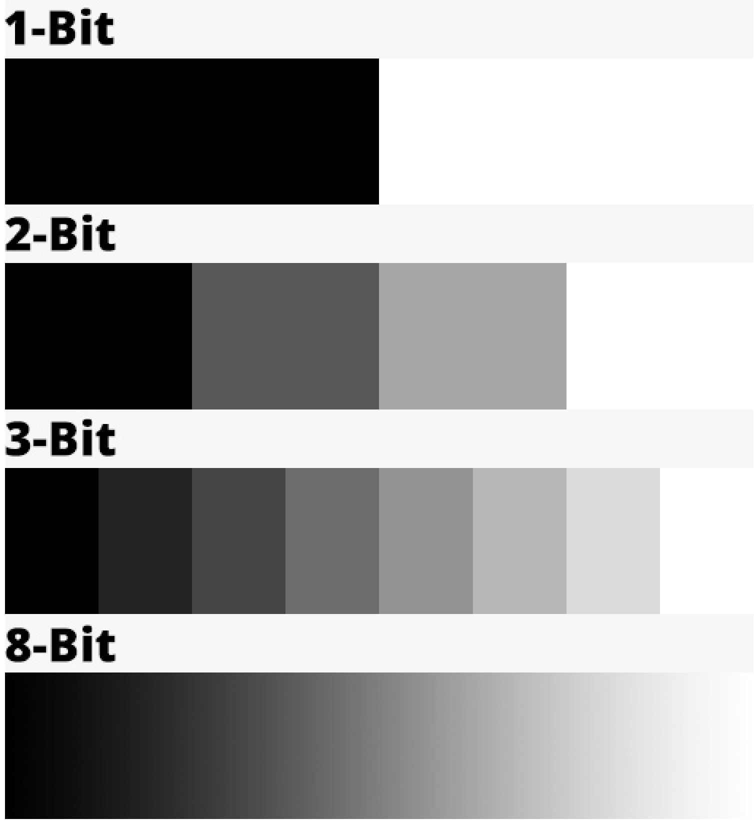
Nguyen Thanh Huy

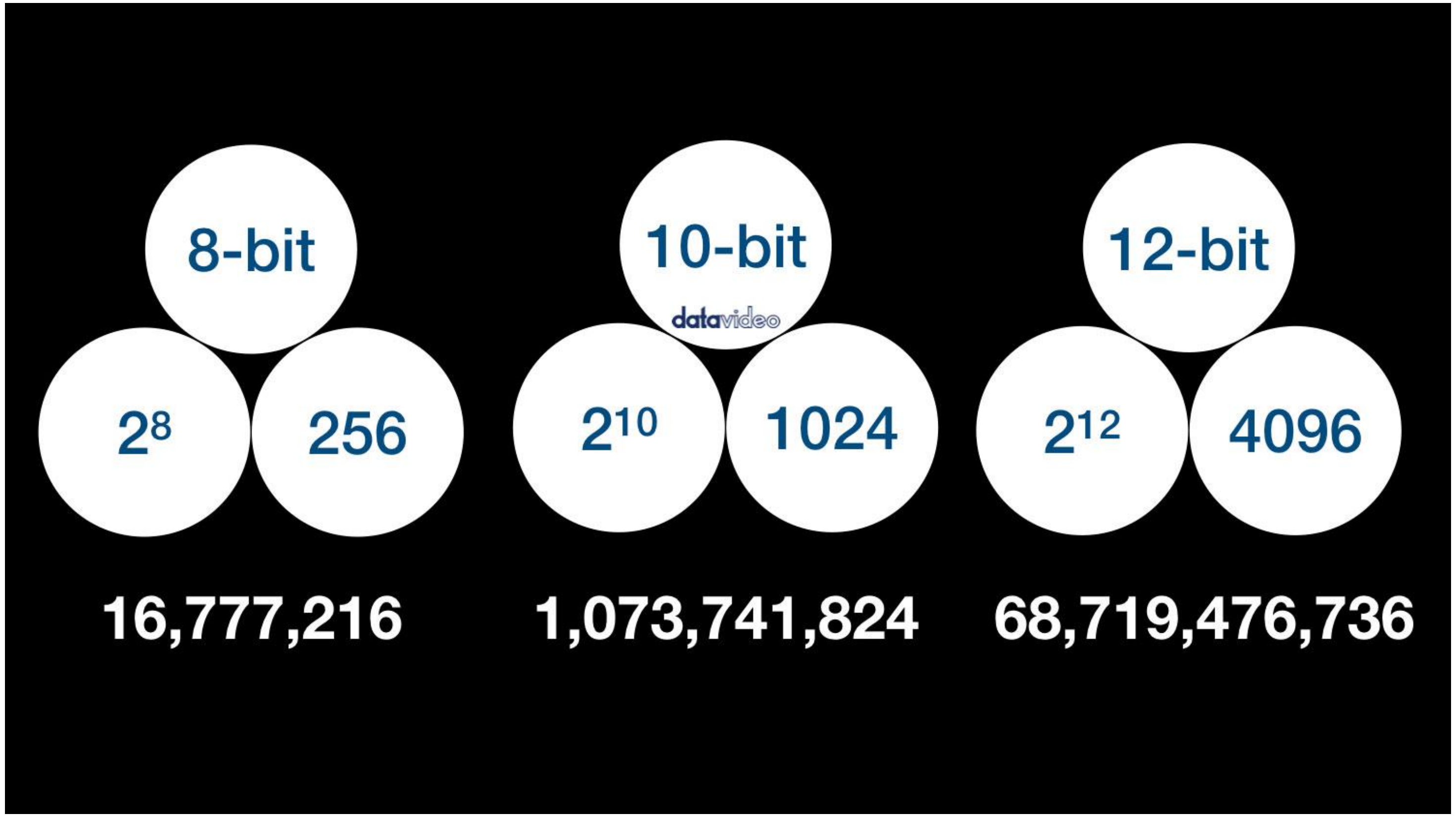
Medical Image Processing



Digital Image Processing (DIP)

❖ N-Bit Image

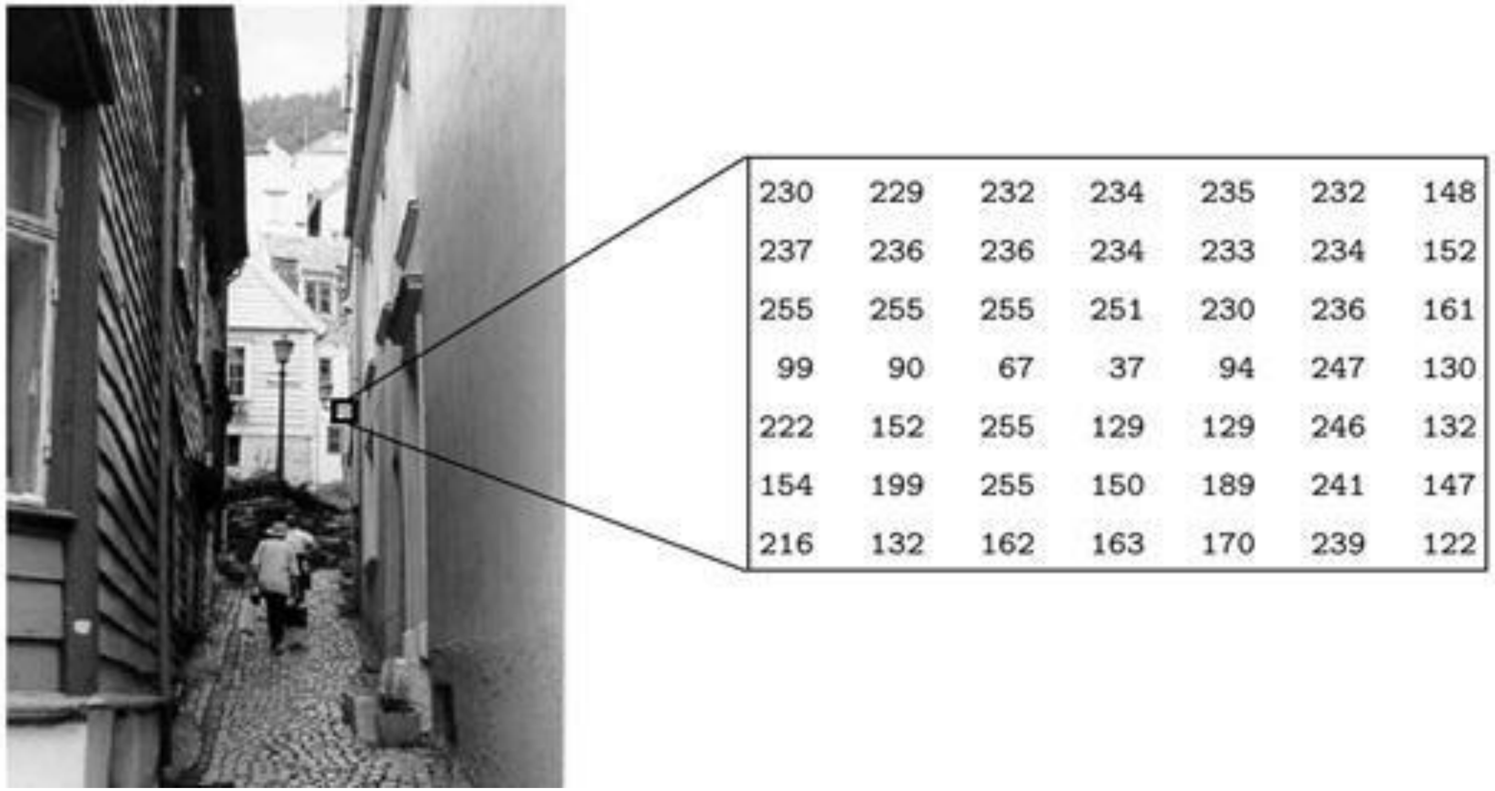




Digital Image Processing (DIP)

❖ 8-Bit Image

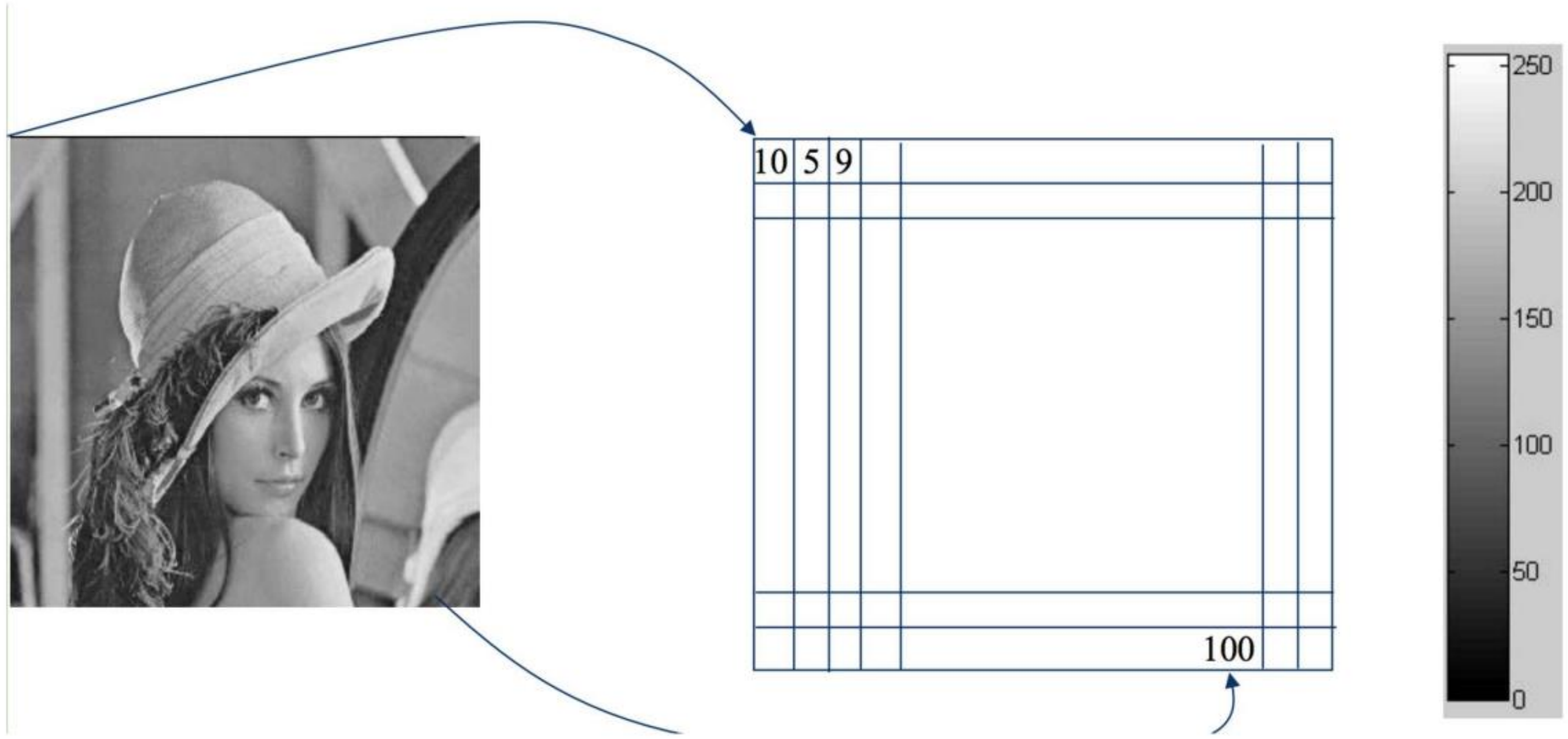
- ❑ 8-bit is the most popular format for human vision and commercial purpose.
- ❑ Most of Computer Vision tasks are all about 8-bit image (except 12-bit in DICOM medical images and some specialized tasks).



230	229	232	234	235	232	148
237	236	236	234	233	234	152
255	255	255	251	230	236	161
99	90	67	37	94	247	130
222	152	255	129	129	246	132
154	199	255	150	189	241	147
216	132	162	163	170	239	122

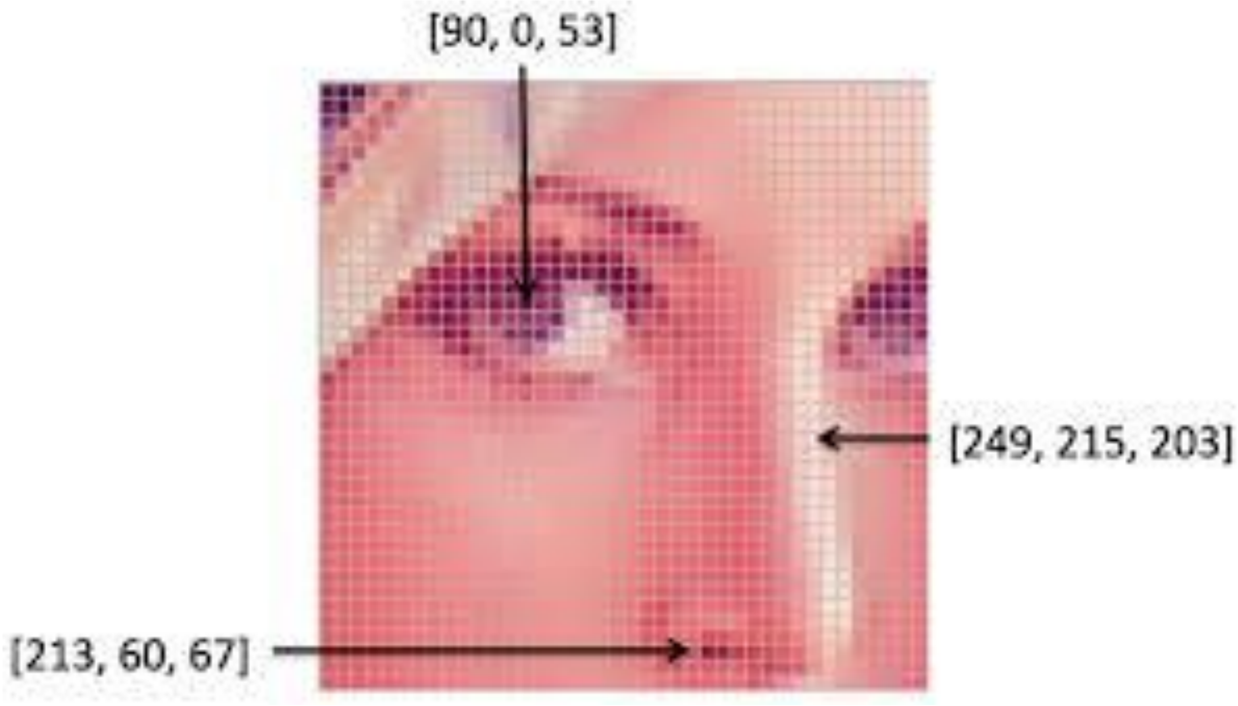
Digital Image Processing (DIP)

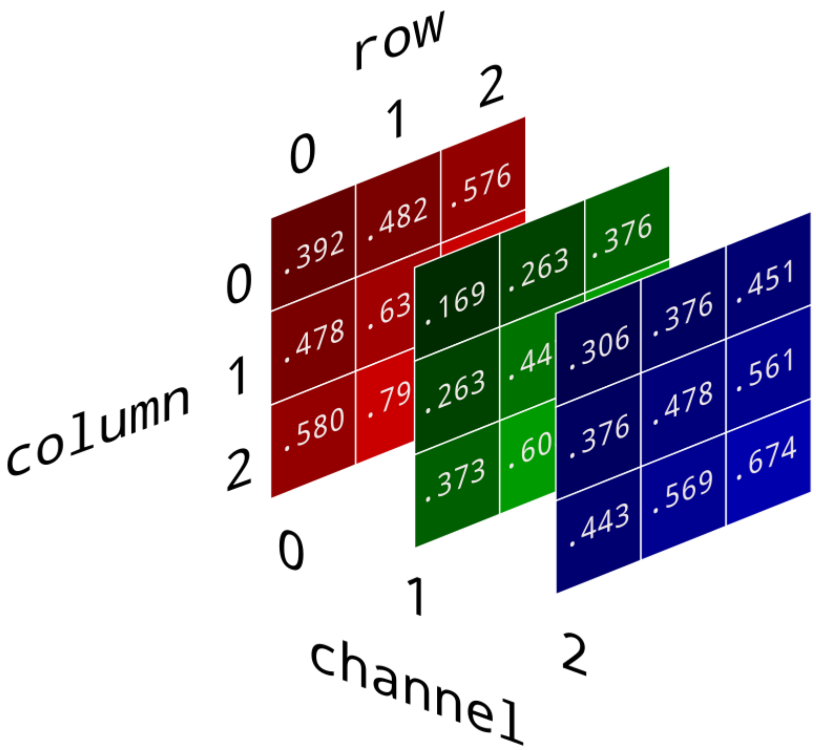
❖ Image Representation by pixels



Digital Image Processing (DIP)

❖ **Image Representation by pixels**

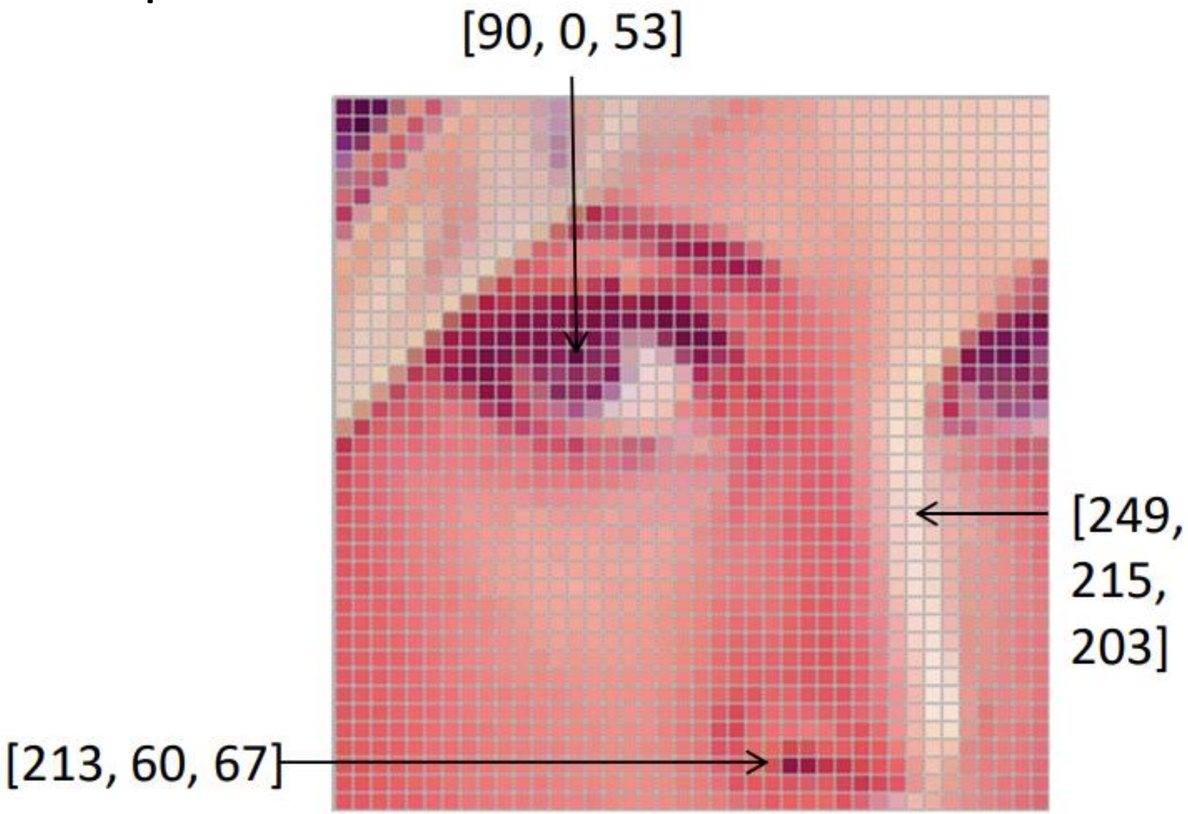




Digital Image Processing (DIP)

❖ Image Representation by pixels

- ❑ Digital image contains discrete pixels.
- ❑ Pixel values:
  - – "grayscale"
    (or "intensity"): [0,255]
  - – "color"
    - • RGB: [R, G, B]
    - • Lab: [L, a, b]
    - • HSV: [H, S, V]



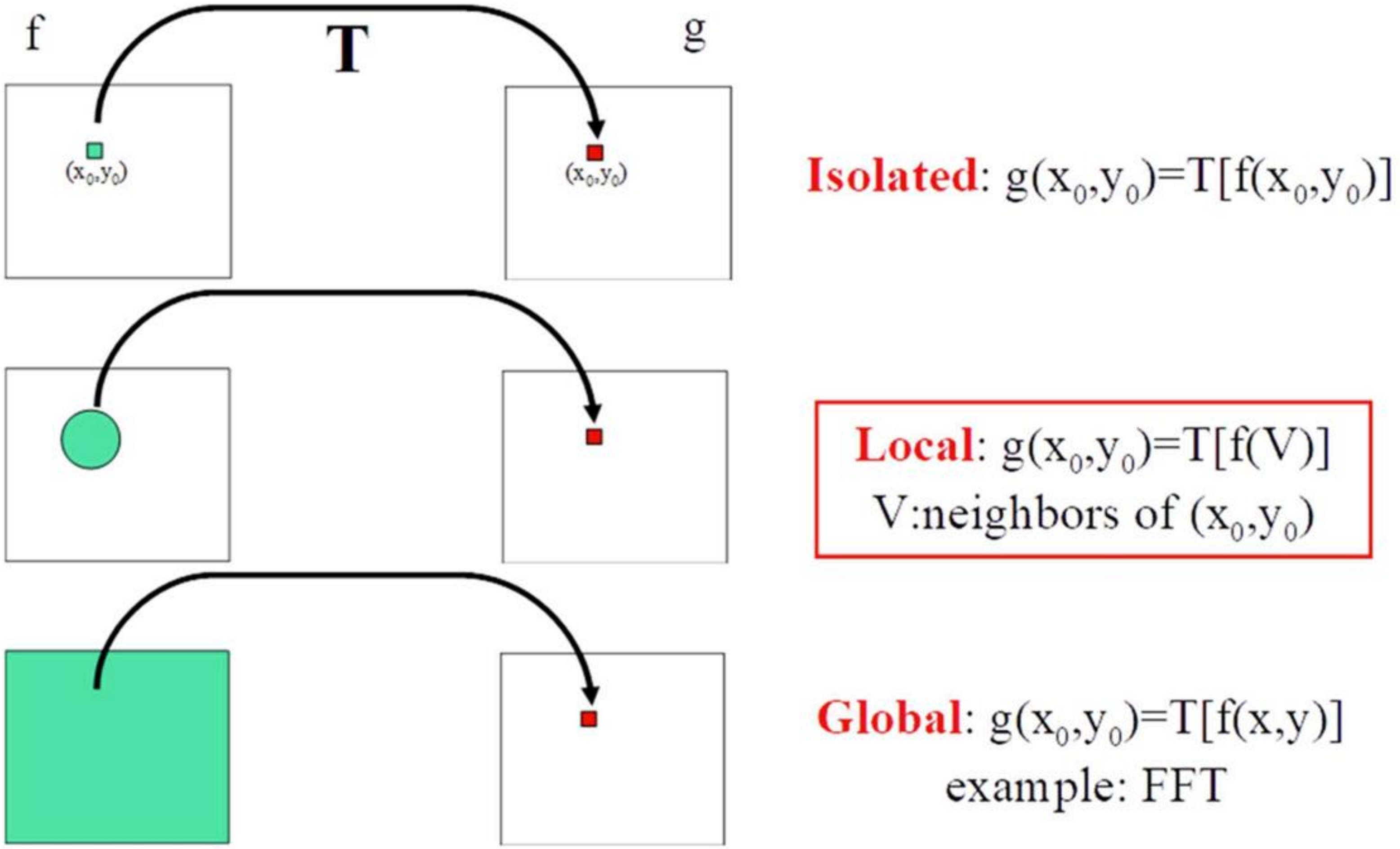
Digital Image Processing (DIP)

❖Pixel value transformation

- Phép biến đổi đơn lẻ (Isolated transformations) - Toán tử điểm của điểm ảnh trên một ảnh
  - *Đọc giá trị một điểm ảnh → thay thế giá trị đó bằng giá trị khác*
  - Ví dụ: tăng cường tương phản, cân bằng histogram
- Phép biến đổi cục bộ (local transformations)
  - *Đọc giá trị nhiều điểm ảnh lân cận → tính toán giá trị mới cho một điểm ảnh*
  - *Nhân chập, tương quan,...*
- ...và phép biến đổi toàn cục (global transformations)
  - *Đọc giá trị tất cả các điểm ảnh của một ảnh → tính toán giá trị mới cho 1 điểm ảnh*
  - *FFT, ...*

Digital Image Processing (DIP)

❖ **Pixel value transformation**



Source : Caroline Rougier. Traitement d'images (IFT2730). Univ. de Montréal.

Digital Image Processing (DIP)

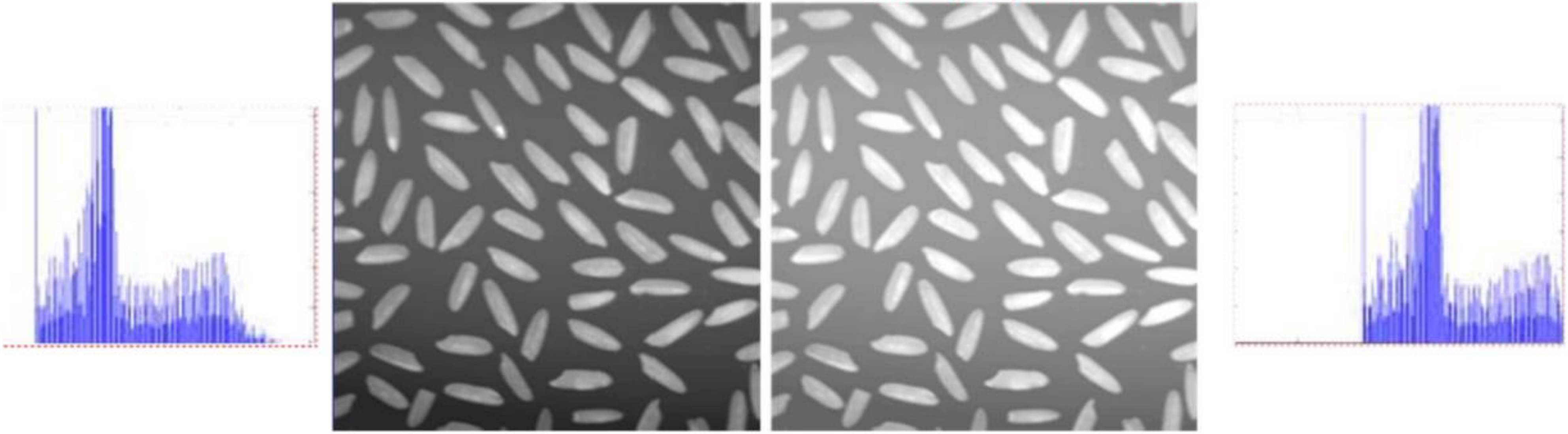
❖ Brightness

- ❑ Average Brightness of all pixels.
- ❑ Represent a light level and dark level of image.

$$B(I) = \frac{1}{wh} \sum_{v=1}^{h} \sum_{u=1}^{w} I(u, v)$$

Divide by total number of pixels

Sum up all pixel intensities



Digital Image Processing (DIP)

❖ Contrast

- Formular:

  - Standard Deviation of all bright values in image.

$$C = \sqrt{\frac{1}{wh}\sum_{u=1}^{h}\sum_{v=1}^{w}(I(u,v) - mean)^2}$$

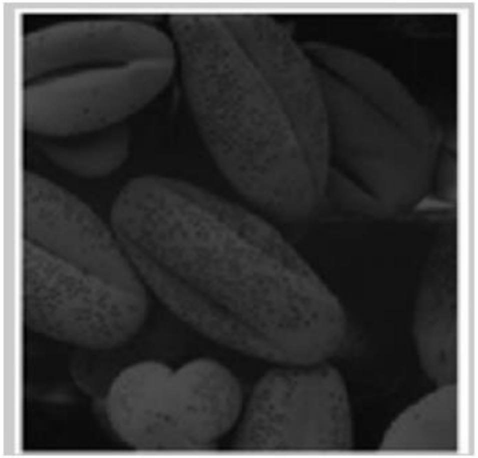
  - The difference between max and min of bright values.

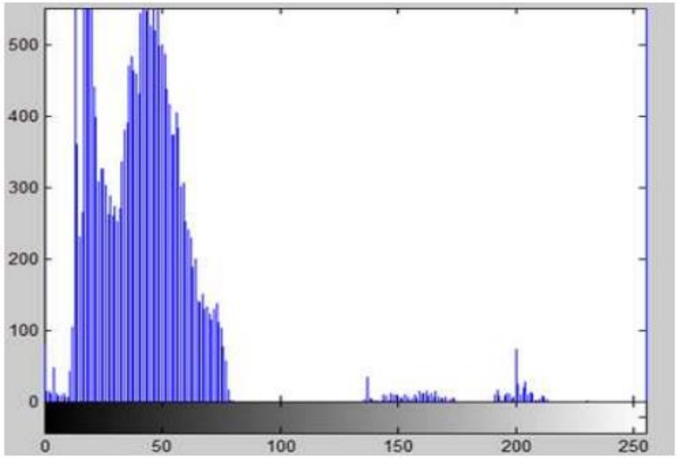
$$C = \frac{\max(I(u,v)) - \min(I(u,v))}{\max(I(u,v)) + \min(I(u,v))}$$

Digital Image Processing (DIP)

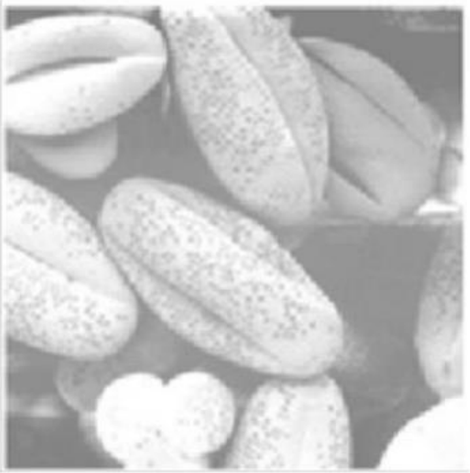
❖Histogram & Brightness

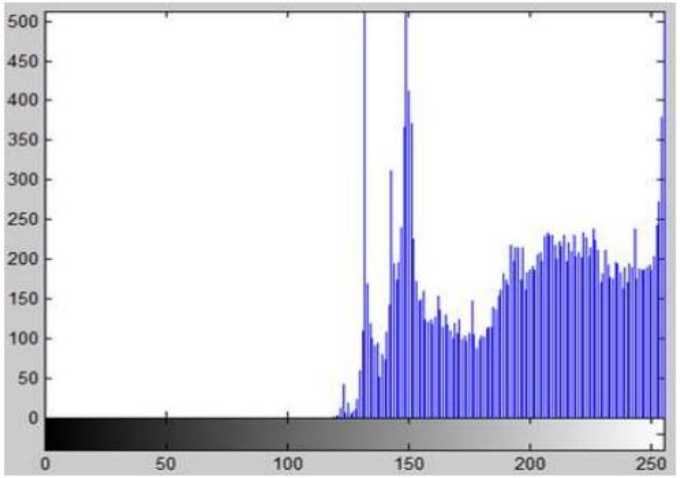
Dark image





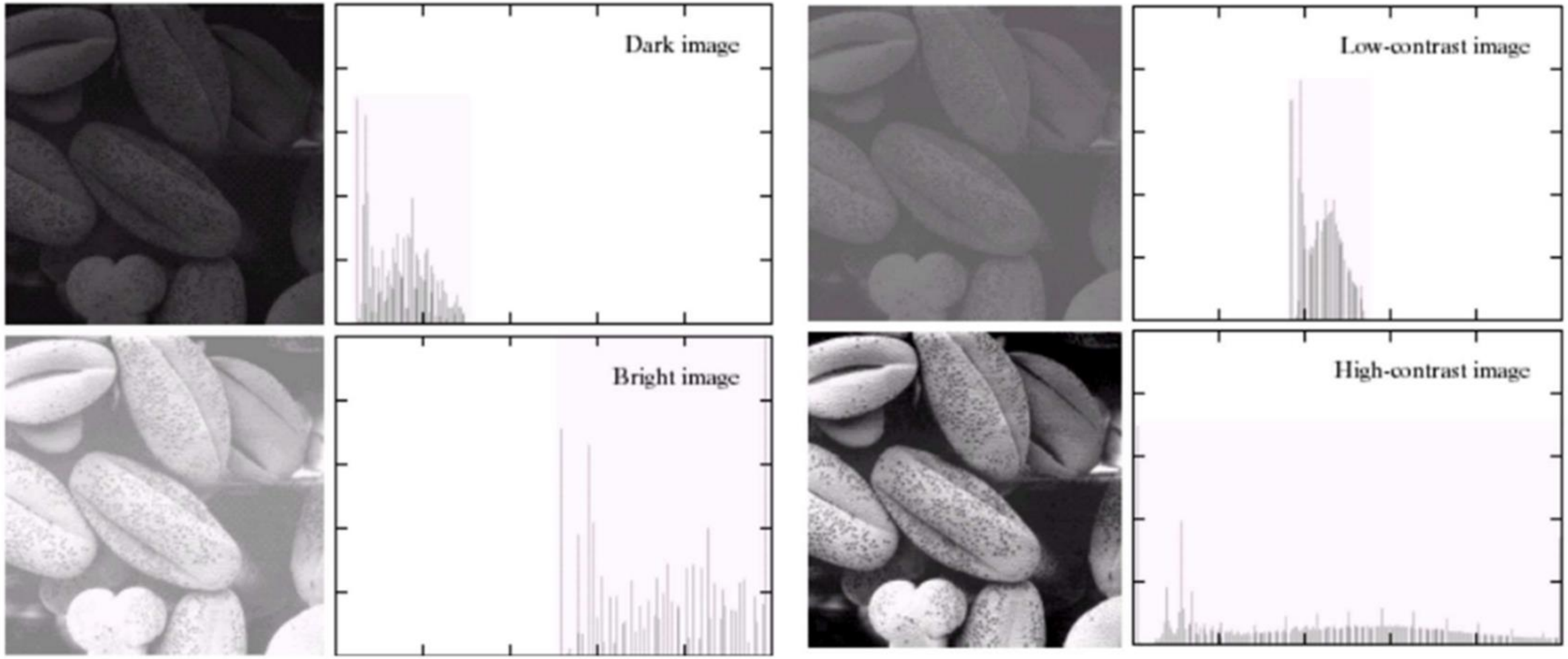
Light image





Digital Image Processing (DIP)

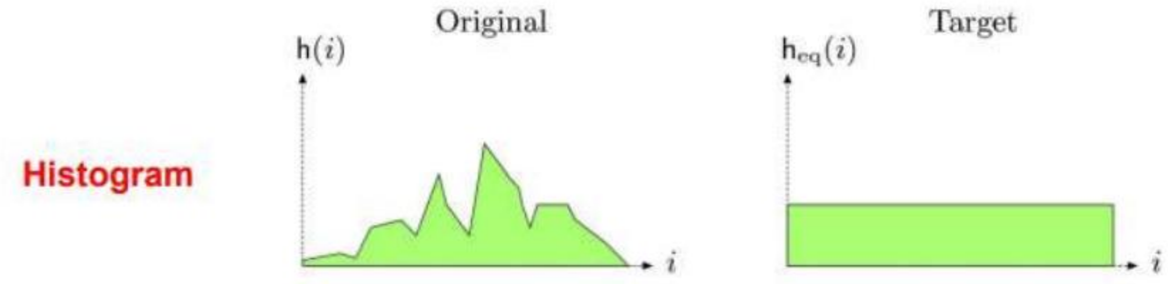
❖ Histogram - Examples



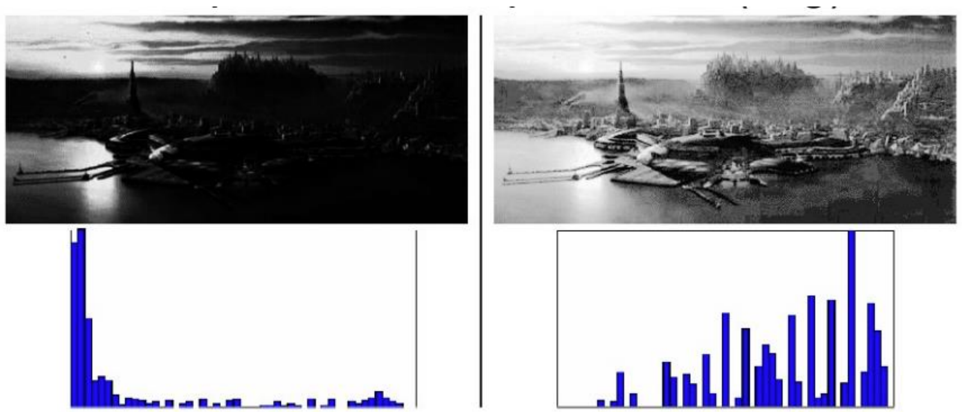
DIP: Contrast Enhancement

❖ Histogram equalization

- ❑ Histogram of image tends to become uniform distribution

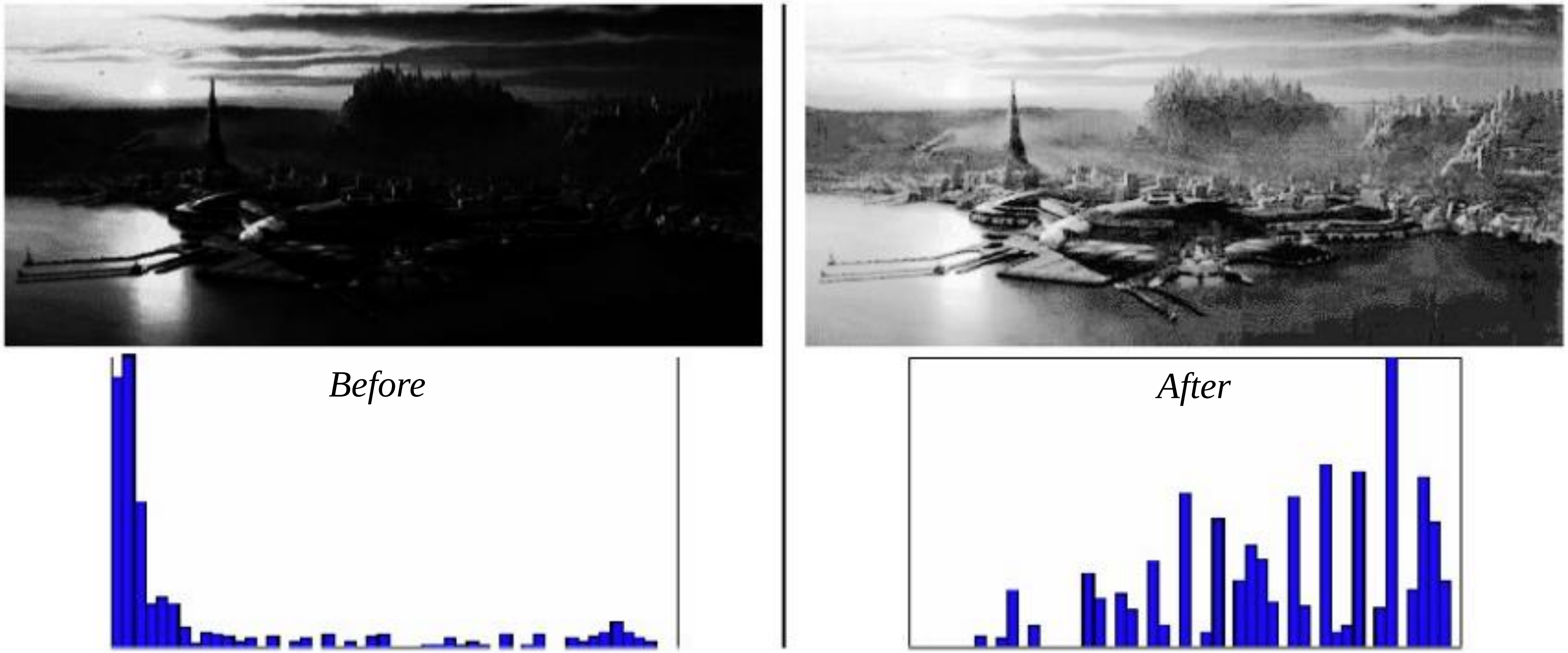


- ❑ Histogram of image tends to become uniform distribution



DIP: Contrast Enhancement

❖ **Histogram equalization**



DIP: Contrast Enhancement

❖ Histogram equalization

- The formula for histogram equalisation in the discrete case is given by

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j$$

Where:

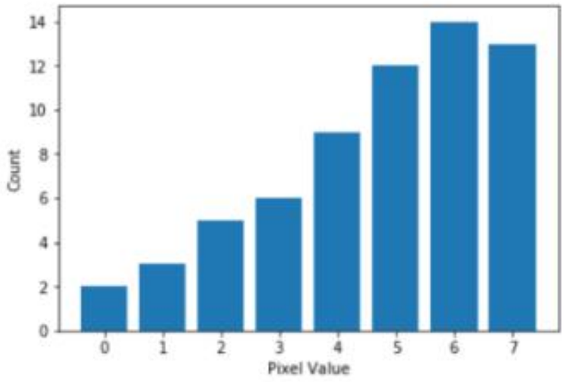
r_k: input intensity

s_k: processed intensity

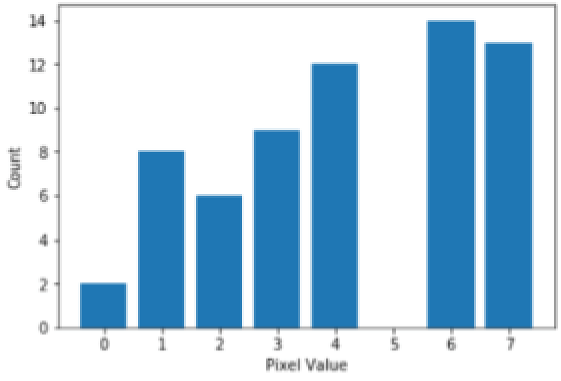
n_j: the frequency of intensity j

MN: number of image pixels

r_k	n_k
0	2
1	3
2	5
3	6
4	9
5	12
6	14
7	13



HistEq

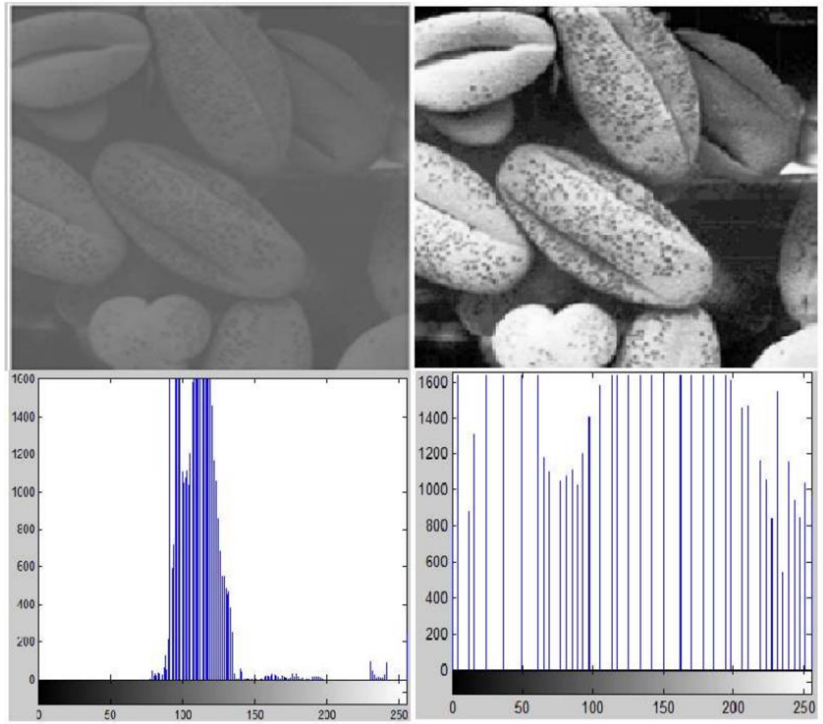


Equalized

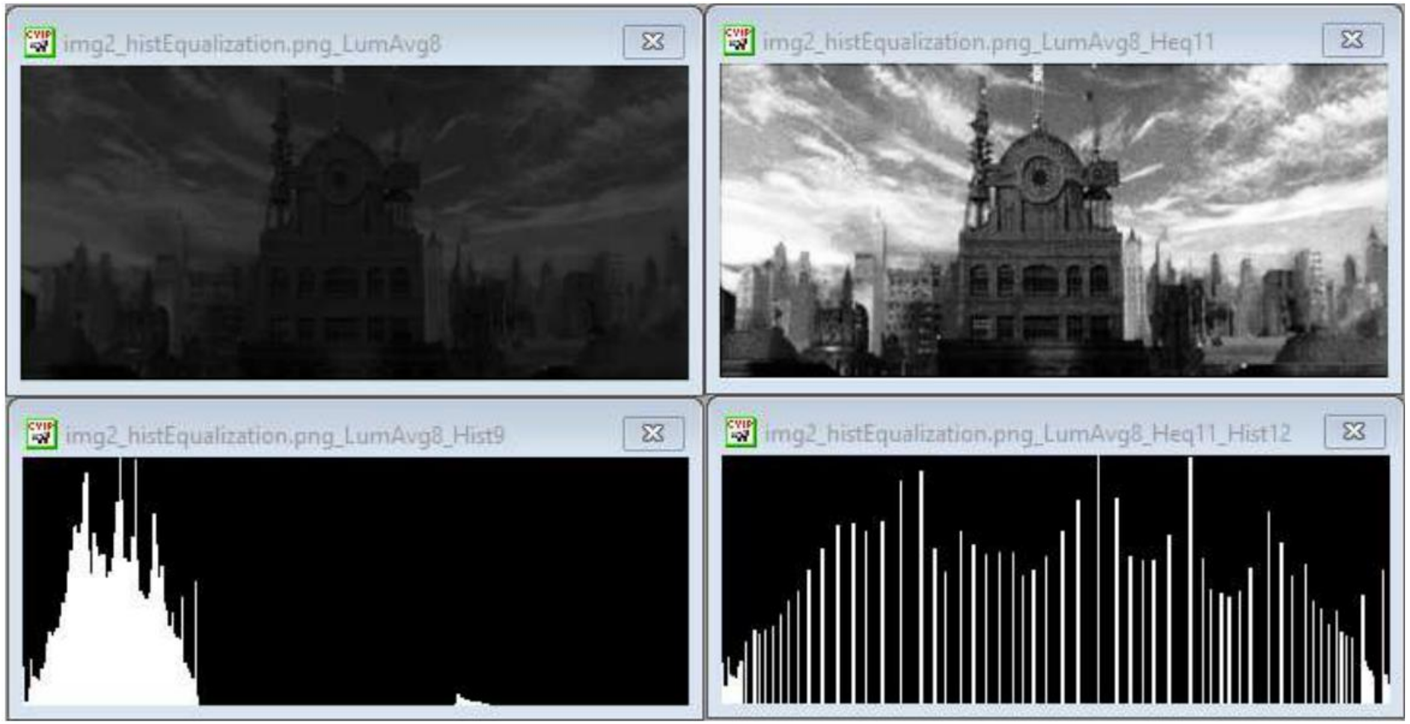
Solution & Reference:
https://theailearner.com/2019/04/01/histogram-equalization/

DIP: Contrast Enhancement

❖Histogram equalization



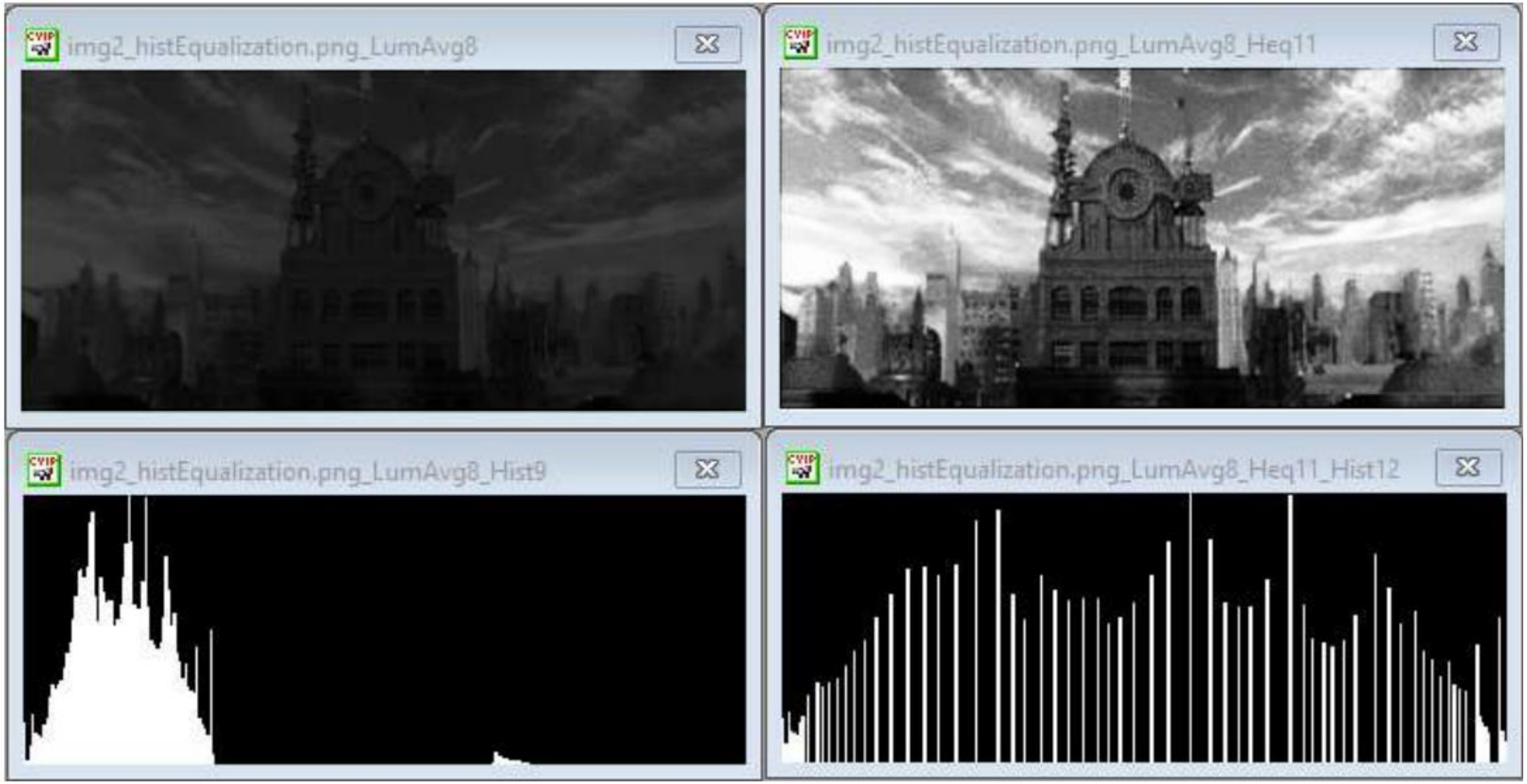
Example 1



Example 2

DIP: Contrast Enhancement

❖Histogram equalization



DIP: Contrast Enhancement

❖Contrast Limited Adaptive Histogram Equalization (CLAHE)

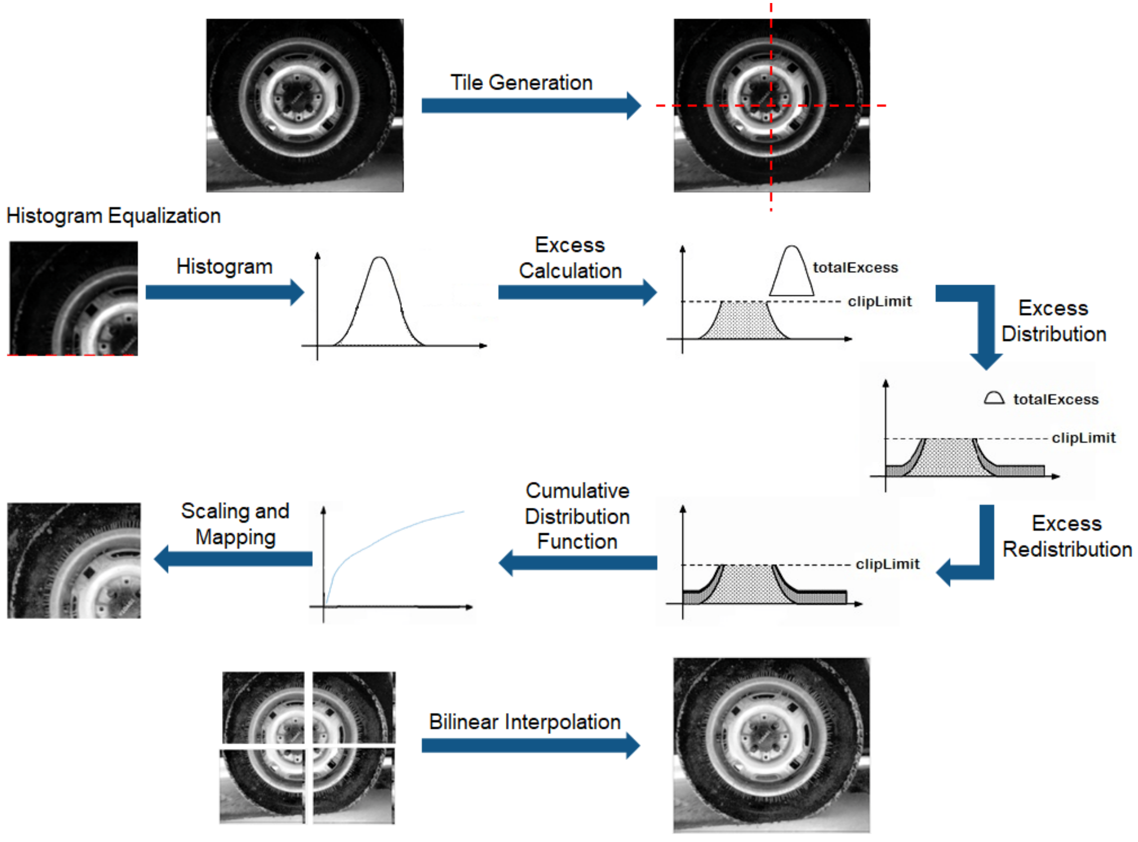


Fig1. Algorithm

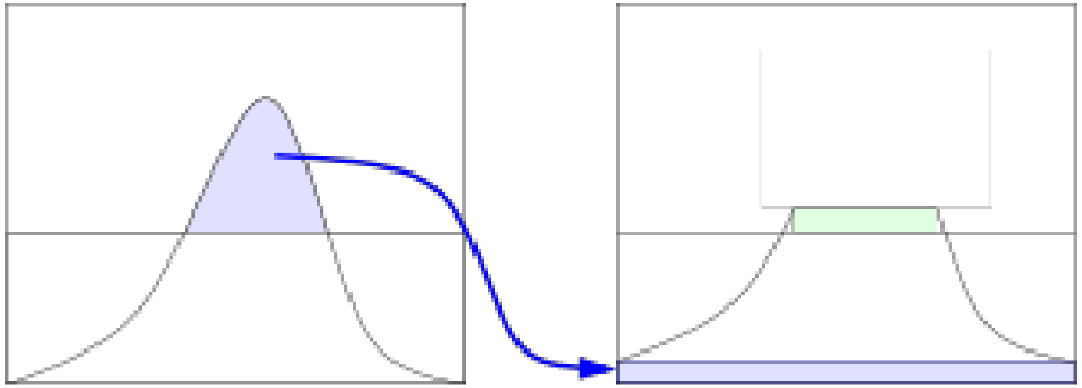
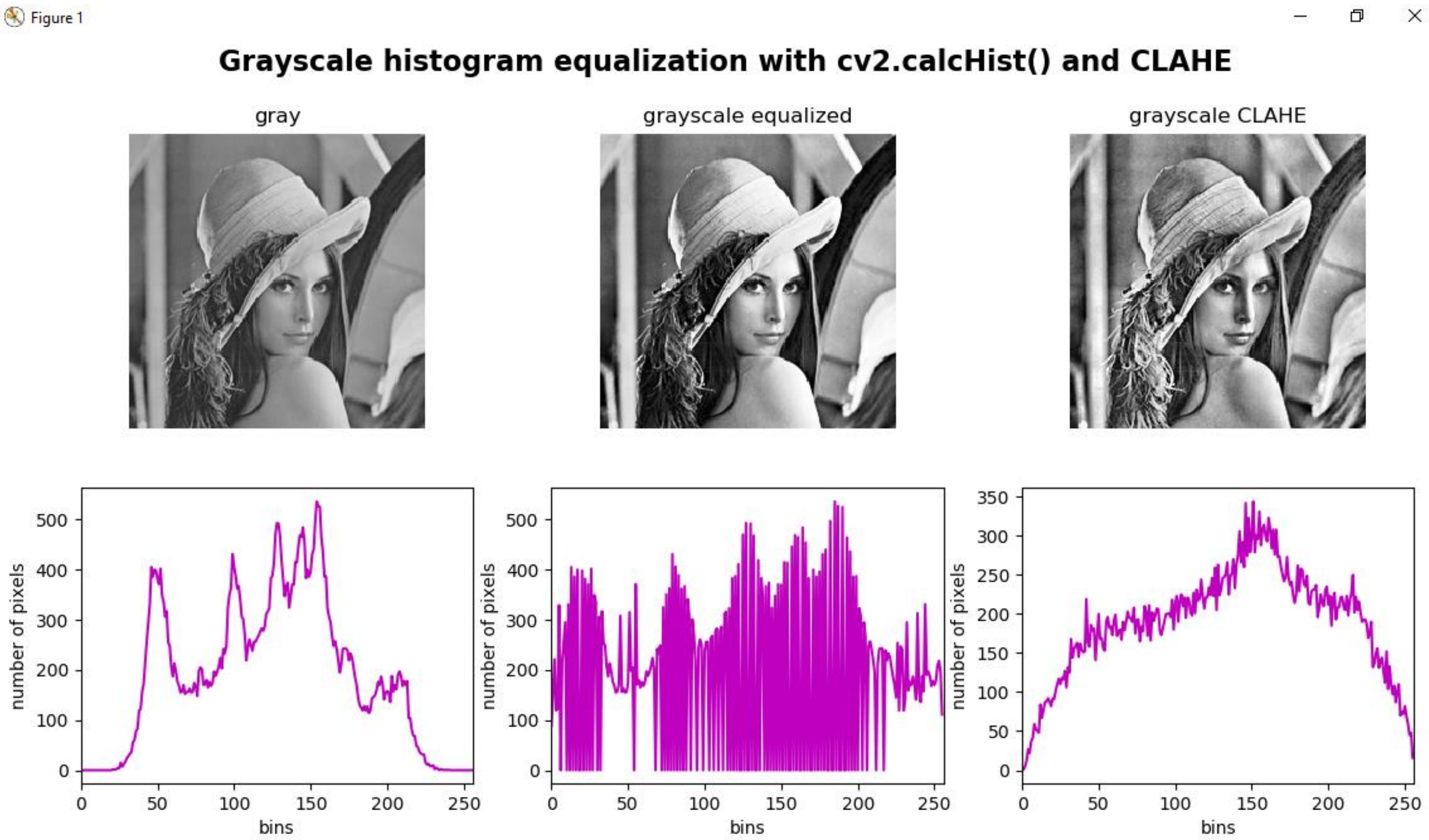


Fig2. Histogram-based concept

Fig3. Visualization

DIP: Contrast Enhancement

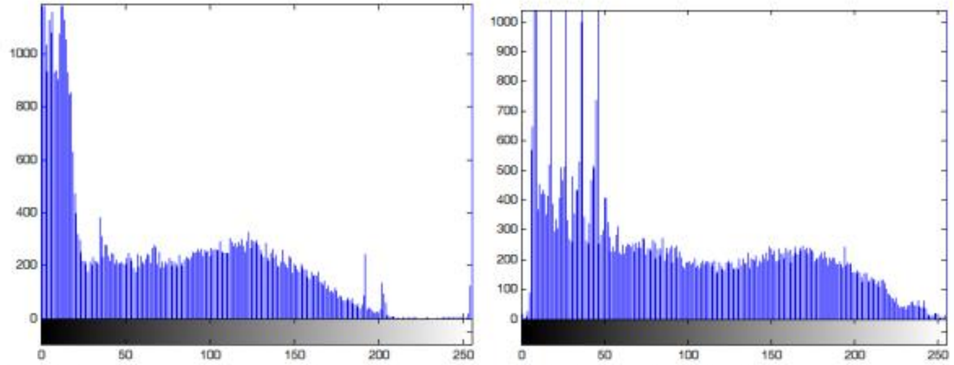
❖Comparison: HE vs CLAHE



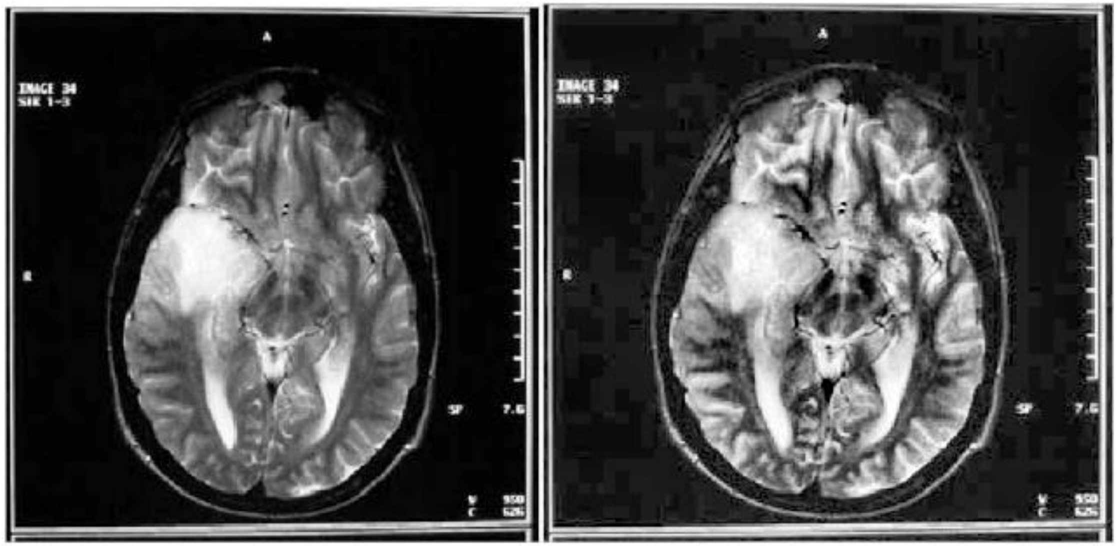
Medical Image Processing

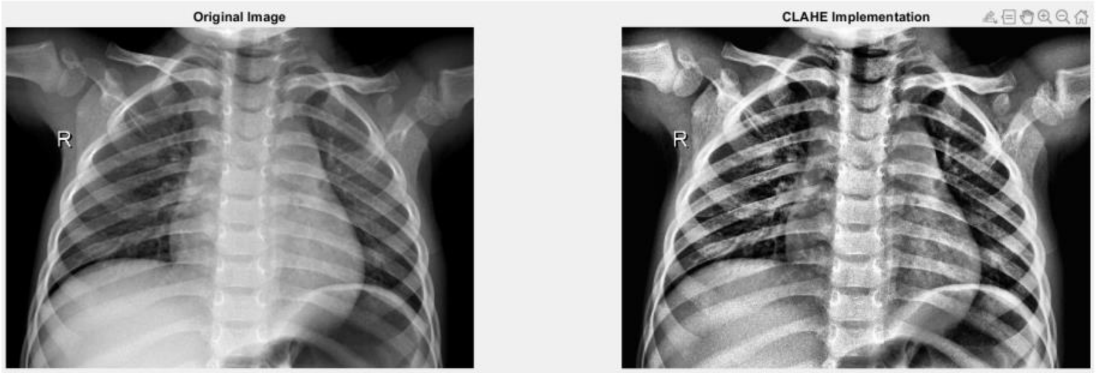
❖ Example

a) Original X_Ray imae 1 b) CLAHE Enhanced image

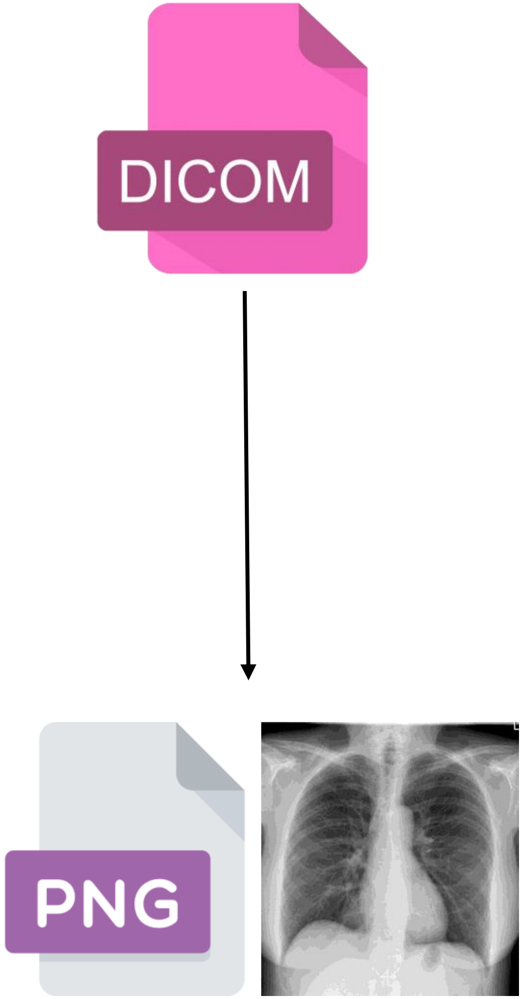


c) Histogram of original image d) Equalized histogram with CLAHE





Medical Image Processing: Demo



❖ **Library**

```python
import os
import pandas as pd
import numpy as np
import cv2
import matplotlib.pyplot as plt
import tensorflow as tf
import tensorflow_io as tfio
```

❖ **Preprocess DICOM file**

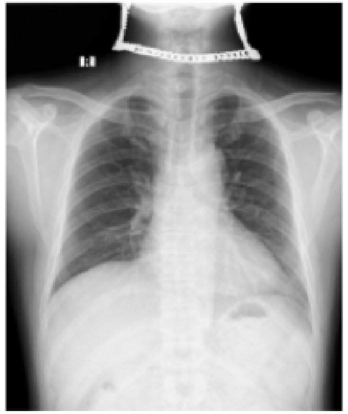
```python
# Reading DICOM images

def read_dicom(path):
    image_bytes = tf.io.read_file(path)
    image = tfio.image.decode_dicom_image(
        image_bytes,
        dtype = tf.uint16
    )

    image = tf.squeeze(image, axis = 0)

    image = tf.image.resize(
        image,
        (500, 500),
        preserve_aspect_ratio = True
    )

    image = image - tf.reduce_min(image)
    image = image / tf.reduce_max(image)
    image = tf.cast(image * 255, tf.uint8)

    return image
```

Medical Image Processing: Demo

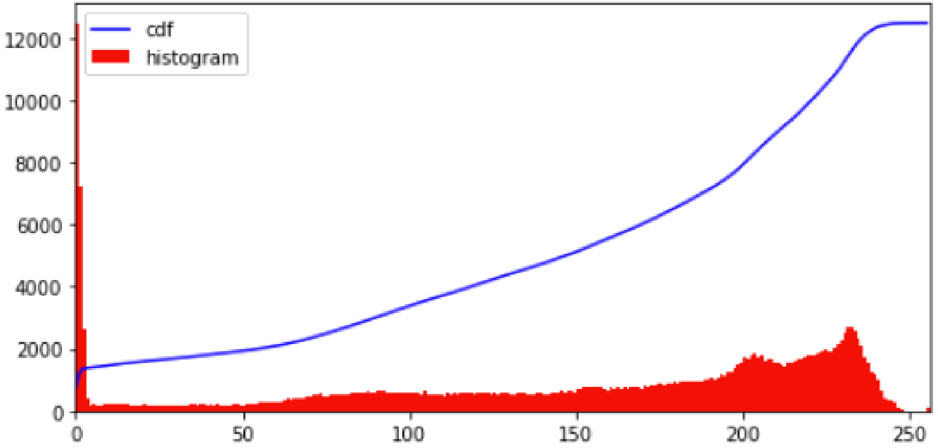
```
fpath = "/kaggle/input/vinbigdata-chest-xray-abnormalities-detection/train/000434271f63a053c4128a0ba6352c7f.dicom"
image = read_dicom(fpath)

plt.figure(figsize = (18, 4))

plt.subplot(1, 2, 1)
plt.axis("off")
plt.title("Original")
plt.imshow(image, cmap = "gray");

plt.subplot(1, 2, 2)
hist, bins = np.histogram(image.numpy().flatten(), 256, [0, 256])
cdf = hist.cumsum()
cdf_normalized = cdf * float(hist.max()) / cdf.max()
plt.plot(cdf_normalized, color = 'b')

plt.hist(image.numpy().flatten(), 256, [0, 256], color = 'r')
plt.xlim([0, 256])
plt.legend(('cdf', 'histogram'), loc = 'upper left')
plt.title("Original")
plt.show();
```

❖ **Original**

Original



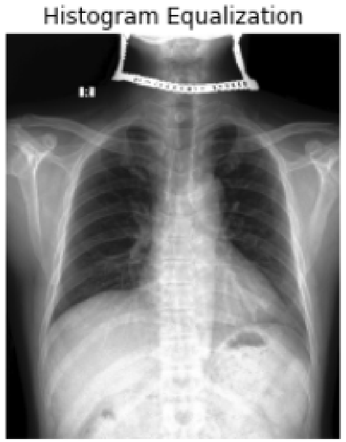
Original

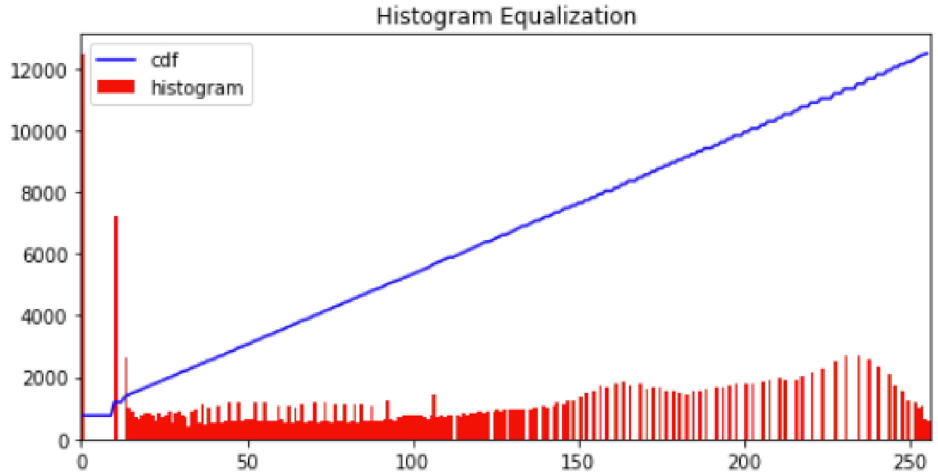


Medical Image Processing: Demo

```
plt.figure(figsize = (18, 4))

plt.subplot(1, 2, 1)
plt.axis("off")
plt.title("Histogram Equalization")

img_he = cv2.equalizeHist(image.numpy())

plt.imshow(img_he, cmap = "gray");

plt.subplot(1, 2, 2)
hist, bins = np.histogram(img_he.flatten(), 256, [0, 256])
cdf = hist.cumsum()
cdf_normalized = cdf * float(hist.max()) / cdf.max()
plt.plot(cdf_normalized, color = 'b')

plt.hist(img_he.flatten(), 256, [0, 256], color = 'r')
plt.xlim([0, 256])
plt.legend(('cdf', 'histogram'), loc = 'upper left')
plt.title("Histogram Equalization")
plt.show();
```

❖ **Histogram Equalization (HE)**





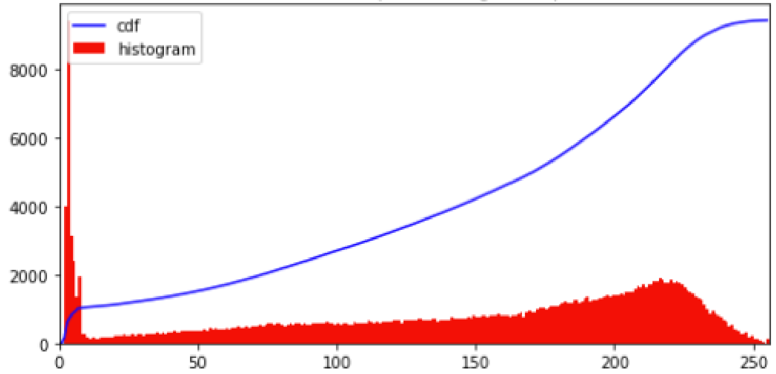
Medical Image Processing: Demo

```
plt.figure(figsize = (18, 4))

plt.subplot(1, 2, 1)
plt.axis("off")
plt.title("Contrast Limited Adaptive Histogram Equalization")

clahe = cv2.createCLAHE(
    clipLimit = 2.,
    tileGridSize = (10, 10)
)

img_clahe = clahe.apply(image.numpy())

plt.imshow(img_clahe, cmap = "gray");

plt.subplot(1, 2, 2)
hist, bins = np.histogram(img_clahe.flatten(), 256, [0, 256])
cdf = hist.cumsum()
cdf_normalized = cdf * float(hist.max()) / cdf.max()
plt.plot(cdf_normalized, color = 'b')

plt.hist(img_clahe.flatten(), 256, [0, 256], color = 'r')
plt.xlim([0, 256])
plt.legend(('cdf', 'histogram'), loc = 'upper left')
plt.title("Contrast Limited Adaptive Histogram Equalization")
plt.show();
```

❖ **CLAHE**

Contrast Limited Adaptive Histogram Equalization

Contrast Limited Adaptive Histogram Equalization



Medical Image Processing: Demo

```
plt.figure(figsize = (18, 4))

plt.subplot(1, 3, 1)
plt.axis("off")
plt.title("Original")
plt.imshow(image, cmap = "gray");

plt.subplot(1, 3, 2)
plt.axis("off")
plt.title("Histogram Equalization")
plt.imshow(img_he, cmap = "gray");

plt.subplot(1, 3, 3)
plt.axis("off")
plt.title("Contrast Limited Adaptive Histogram Equalization")
plt.imshow(img_clahe, cmap = "gray");
```

Medical Image Processing

❖Our previous contribution related to image enhancement

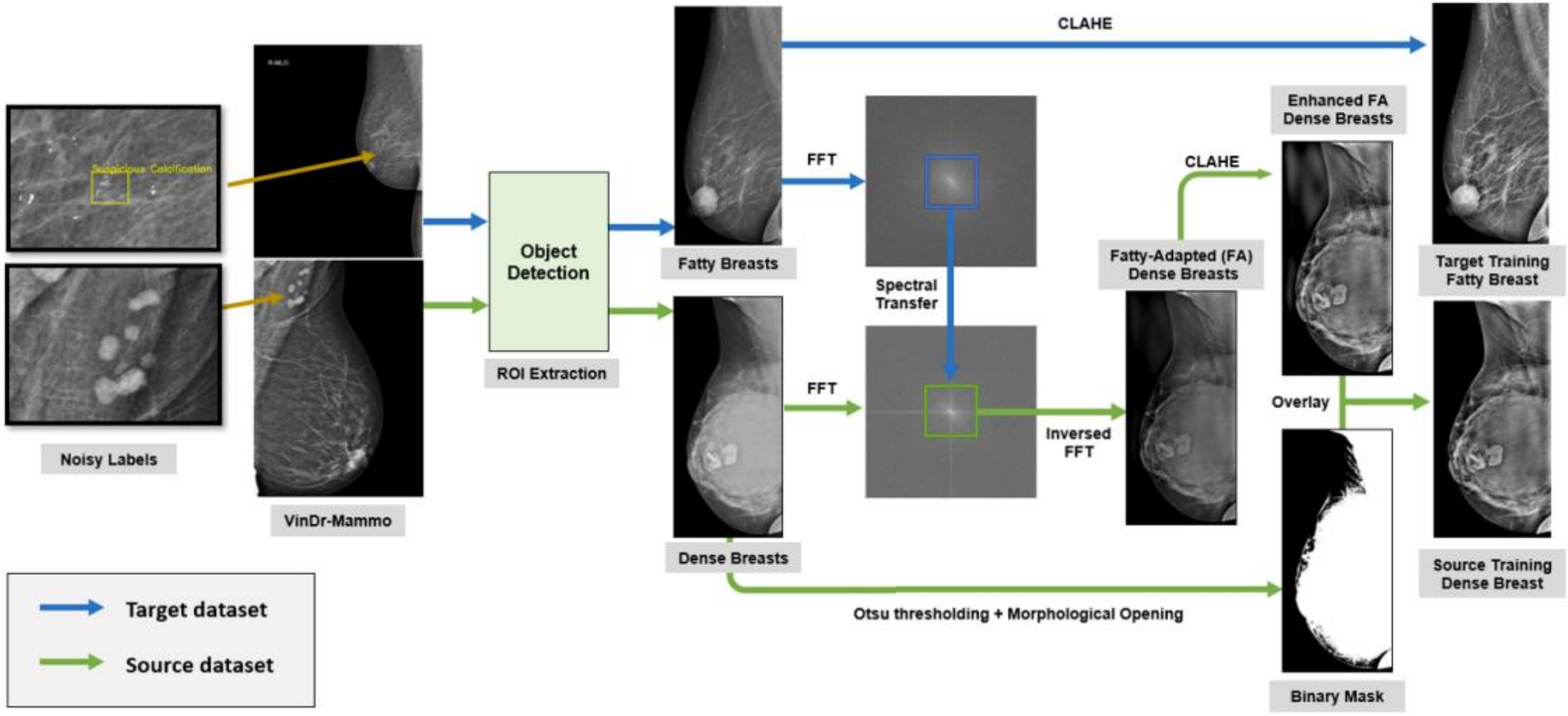


Fig. 1. FALCE: Fourier-adapted Locality Contrast Enhancement framework for preprocessing and creating the fatty-adapted dense images before applying CLAHE under DIP-based created a binary mask for enhancing locality semantic information.

H. T. Nguyen, T. B. Lam, Q. T. D. Tran, M. T. Nguyen, D. T. Chung and V. Q. Dinh, "In-context Cross-Density Adaptation on Noisy Mammogram Abnormalities Detection," 2023 IEEE Statistical Signal Processing Workshop (SSP), Hanoi, Vietnam, 2023, pp. 383-387, doi: 10.1109/SSP53291.2023.10208020.

Deep Learning on Medical Imaging

AI VIET NAM
@aivietnam.edu.vn

Nguyen Thanh Huy

U-Net for Biomedical Image Segmentation

U-Net: Convolutional Networks for Biomedical Image Segmentation

Olaf Ronneberger, Philipp Fischer, and Thomas Brox

Computer Science Department and BIOSS Centre for Biological Signalling Studies,
University of Freiburg, Germany
ronneber@informatik.uni-freiburg.de,
WWW home page: http://lmb.informatik.uni-freiburg.de/

Abstract. There is large consent that successful training of deep networks requires many thousand annotated training samples. In this paper, we present a network and training strategy that relies on the strong use of data augmentation to use the available annotated samples more efficiently. The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. We show that such a network can be trained end-to-end from very few images and outperforms the prior best method (a sliding-window convolutional network) on the ISBI challenge for segmentation of neuronal structures in electron microscopic stacks. Using the same network trained on transmitted light microscopy images (phase contrast and DIC) we won the ISBI cell tracking challenge 2015 in these categories by a large margin. Moreover, the network is fast. Segmentation of a 512x512 image takes less than a second on a recent GPU. The full implementation (based on Caffe) and the trained networks are available at http://lmb.informatik.uni-freiburg.de/people/ronneber/u-net.

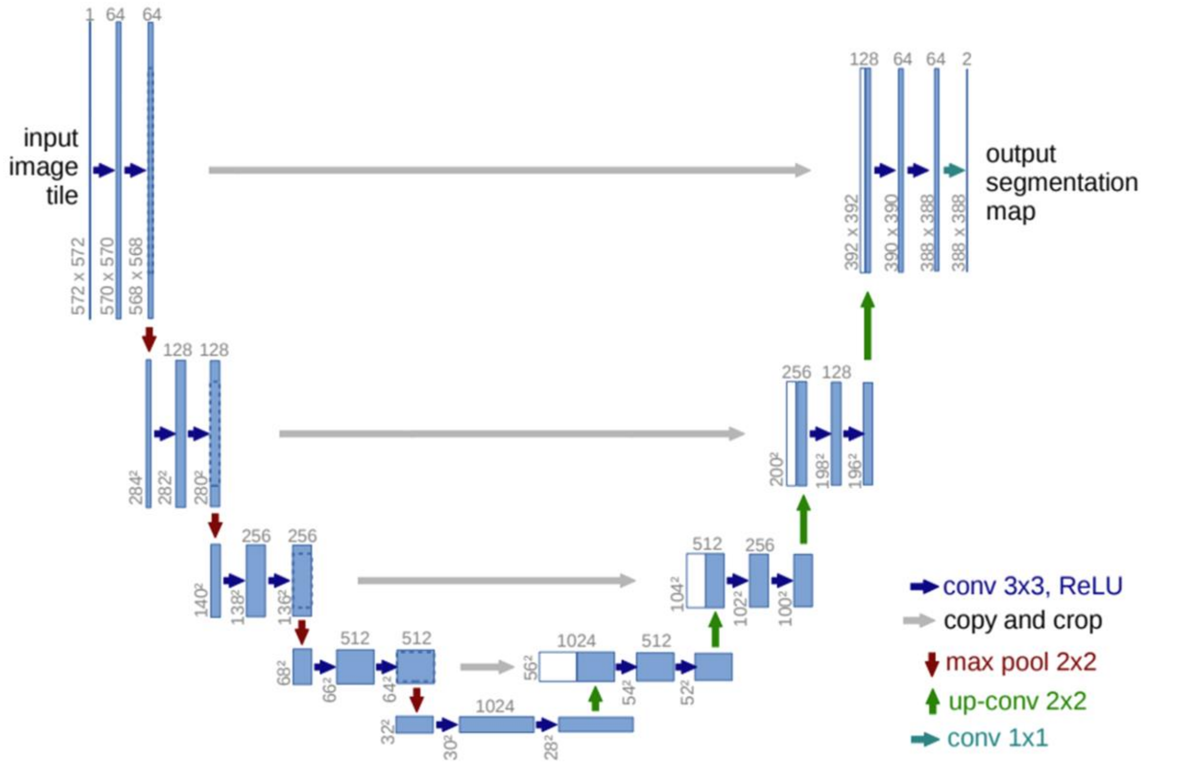
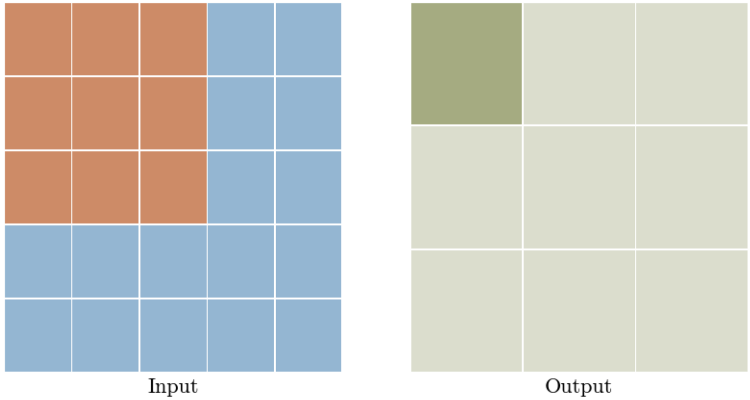


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

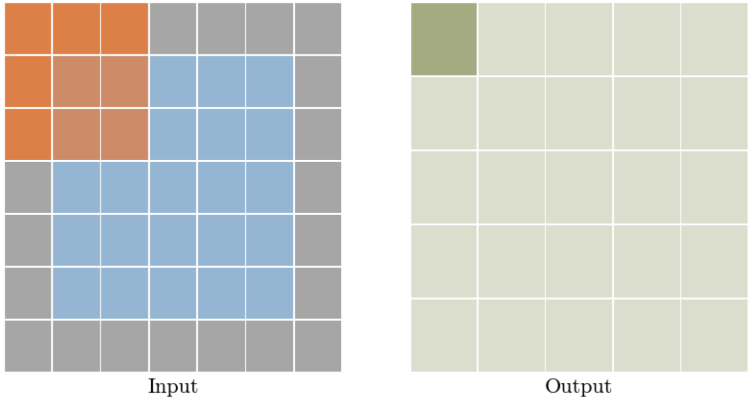
Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 (pp. 234-241). Springer International Publishing.

Standard Convolutional Layers

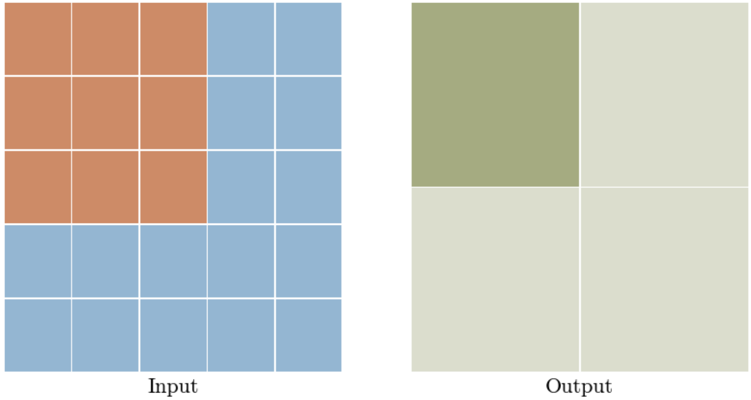
Type: conv - Stride: 1 Padding: 0



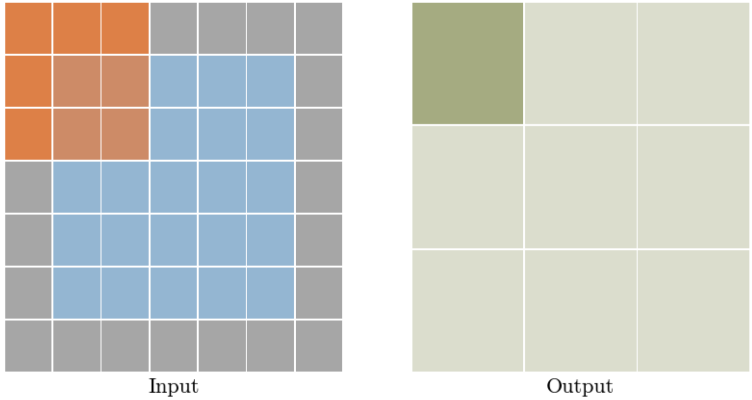
Type: conv - Stride: 1 Padding: 1



Type: conv - Stride: 2 Padding: 0

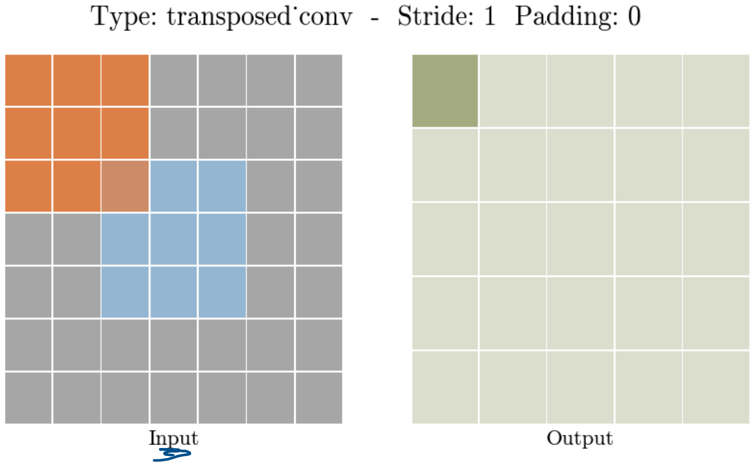


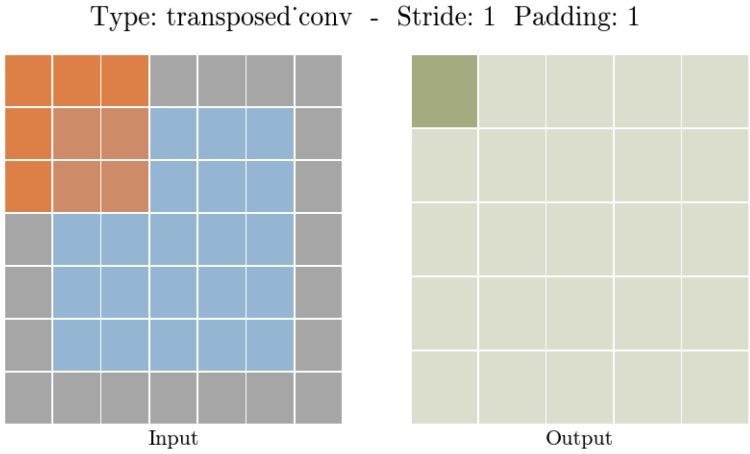
Type: conv - Stride: 2 Padding: 1

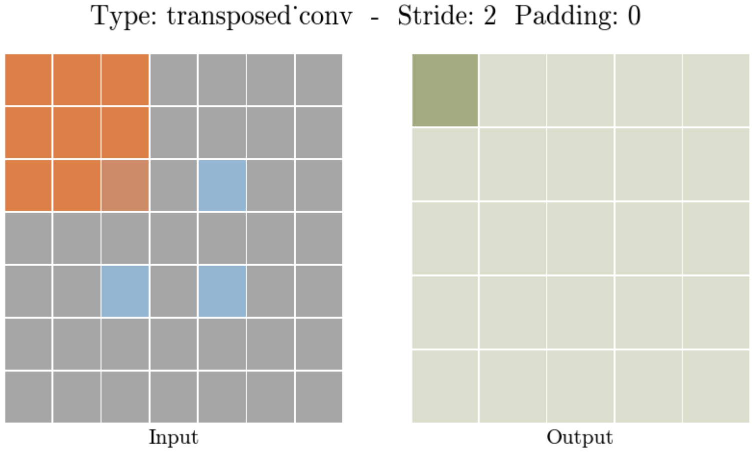


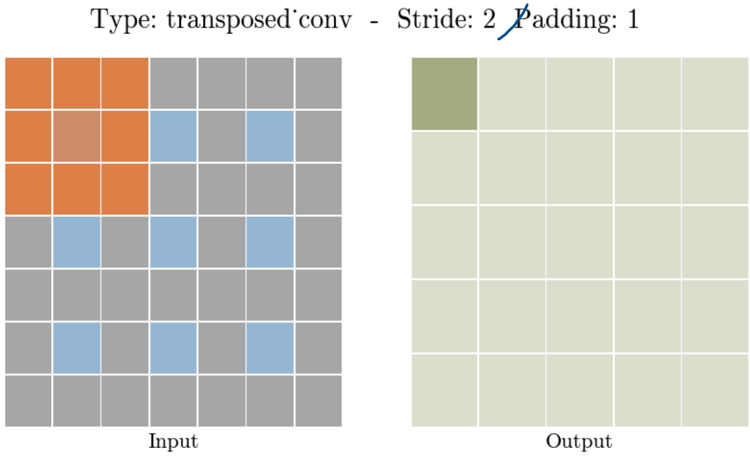
✓ Standard Convolutional Layer does a down-sampling i.e. the spatial dimensions of the output are less than that of the input.

Transposed Convolutional Layers



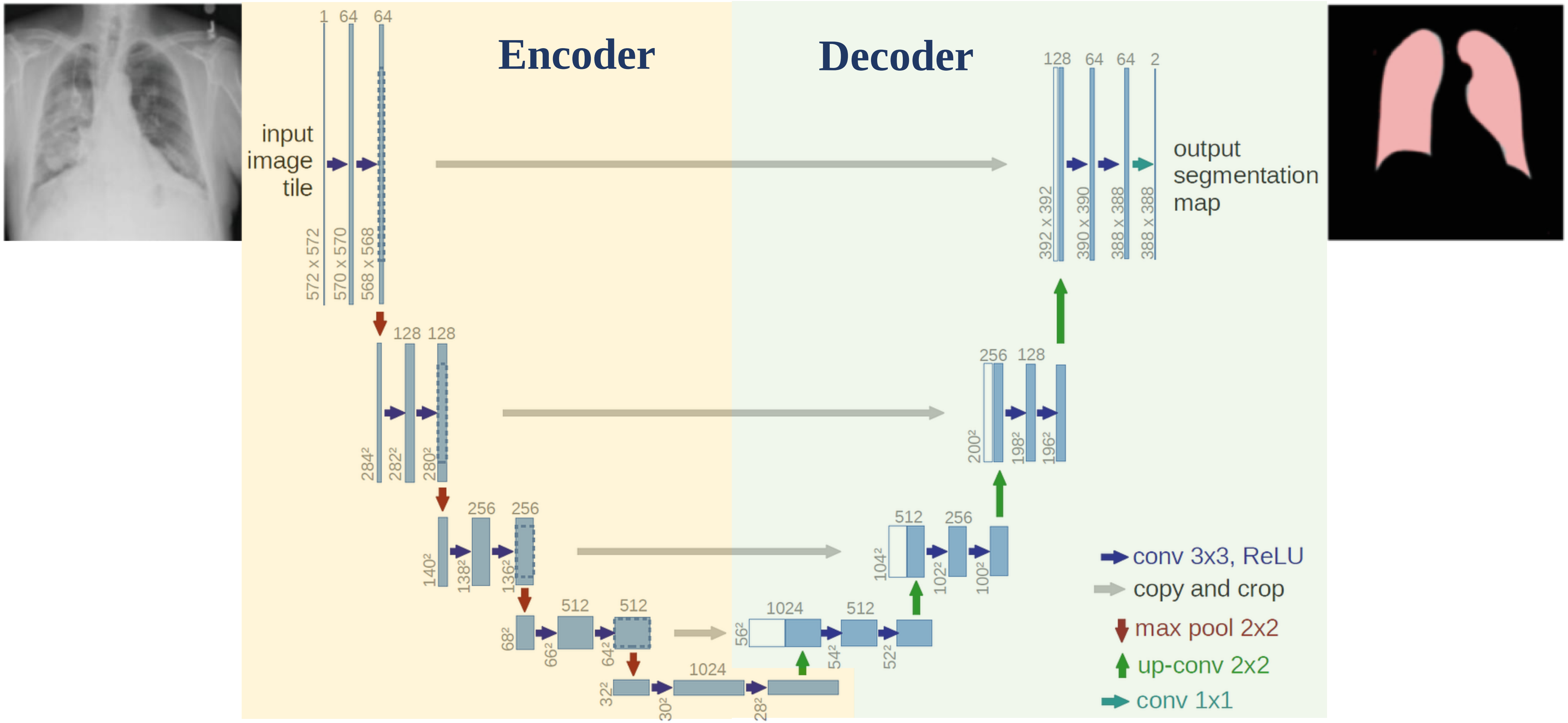






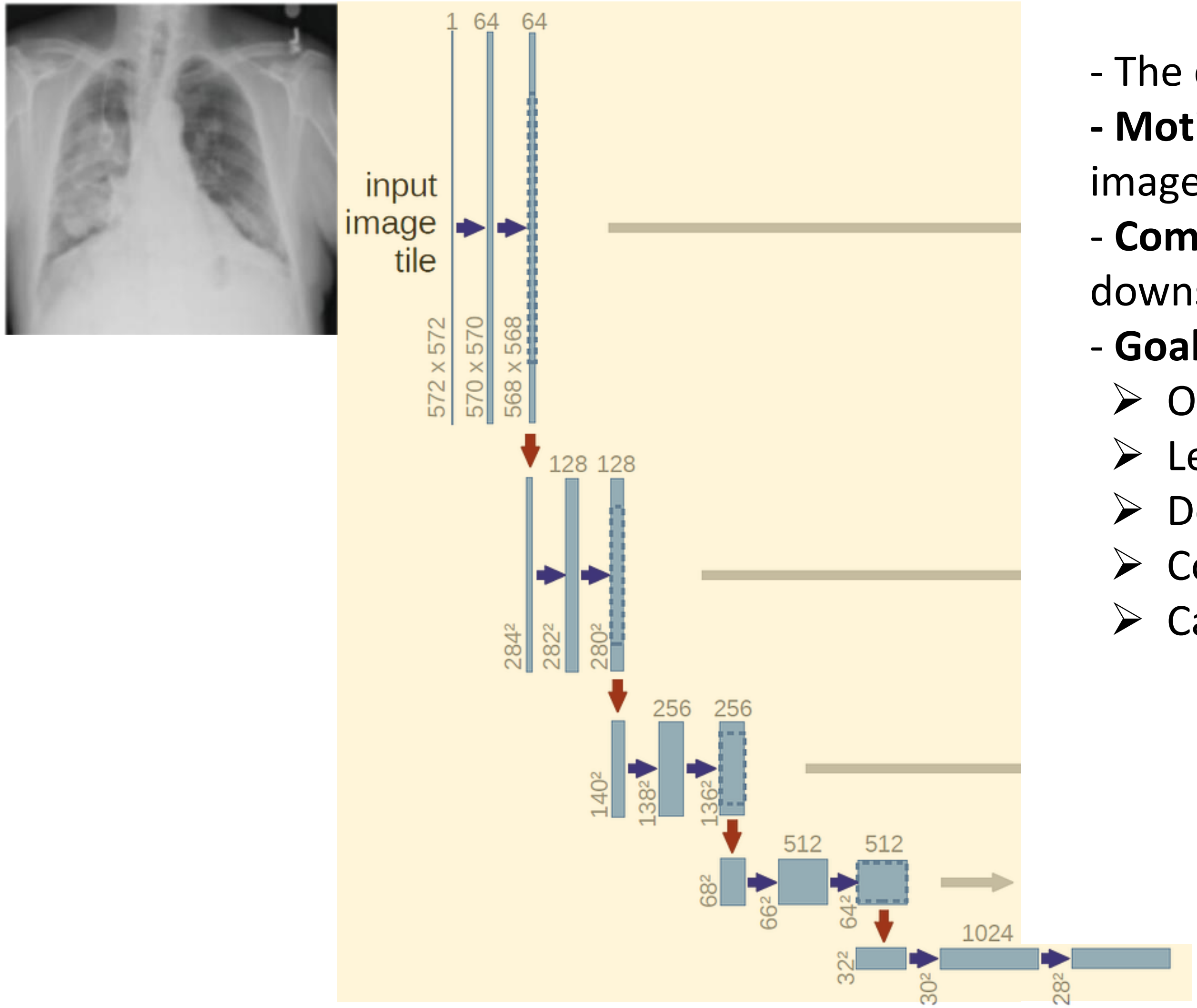
✓ Transposed convolutional layer is carried out for upsampling i.e. to generate an output feature map that has a spatial dimension greater than that of the input feature map.

U-Net for Biomedical Image Segmentation



Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 (pp. 234-241). Springer International Publishing.

Encoder – Contracting Path



- The **encoder** in U-Net is the **contracting path**.

- Motivation: Extract context and compresses the input image by gradually decreasing the spatial dimensions.

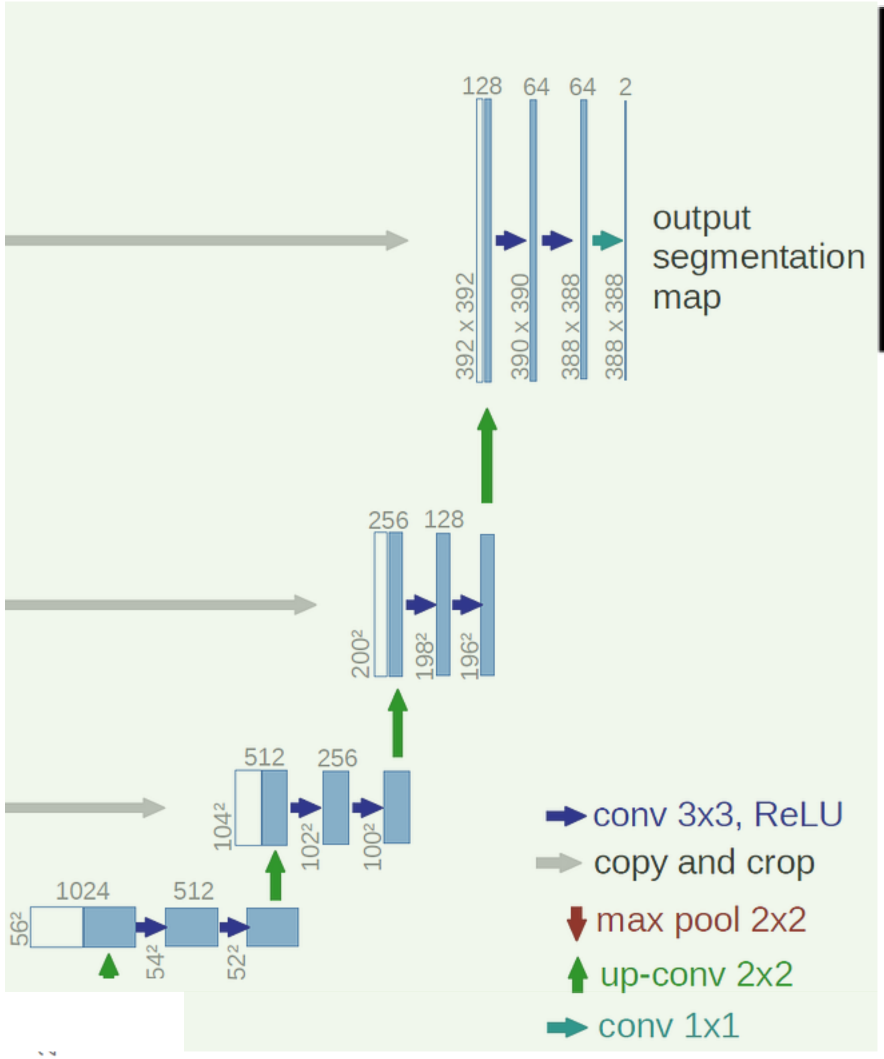
- **Components**: Convolutional Layers + Max Pooling to downsample the feature maps.

- **Goals**:
  - Obtaining high-level characteristics
  - Learning global context
  - Decreasing spatial resolution.
  - Compressing and abstracting the input image
  - Capturing relevant information for segmentation.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 (pp. 234-241). Springer International Publishing.

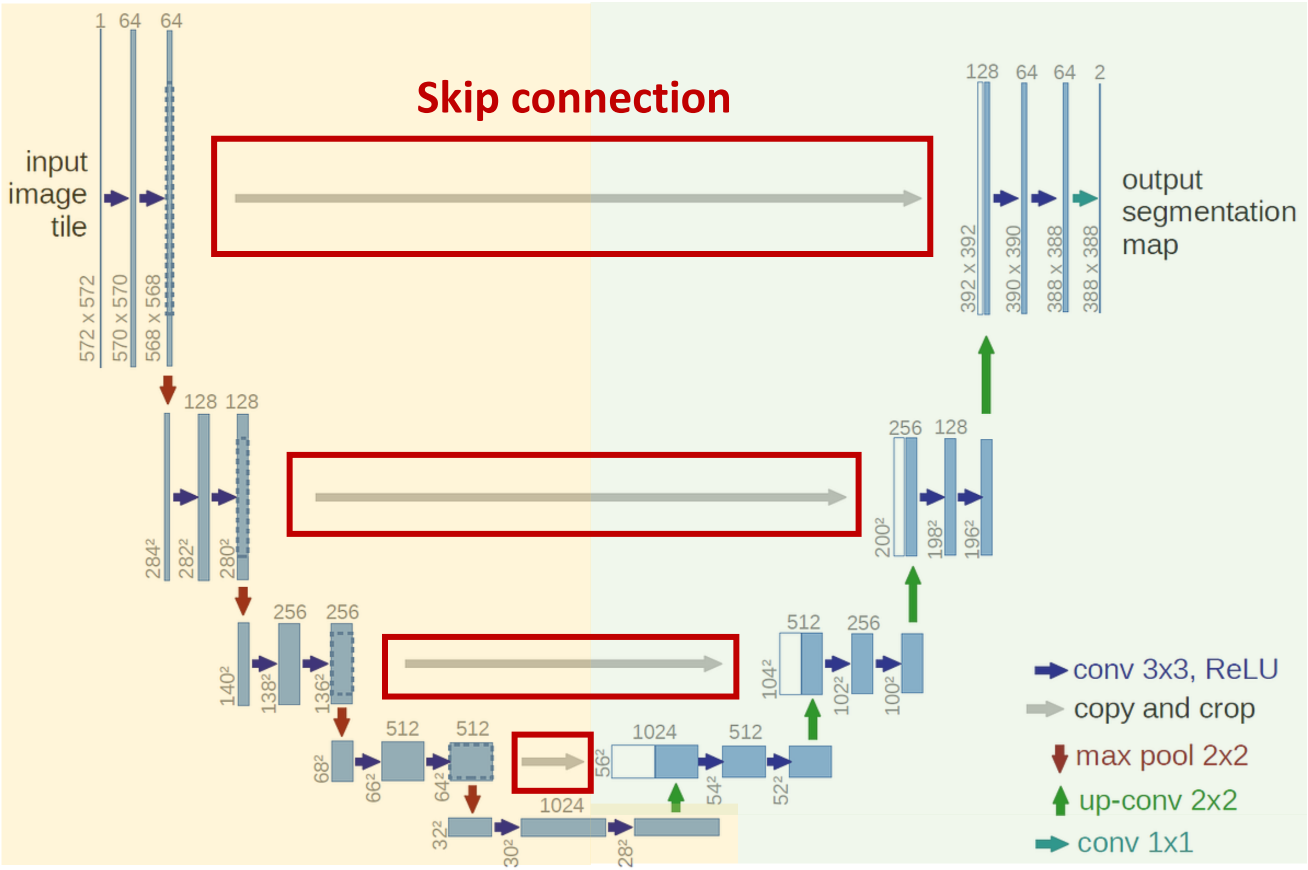
Encoder – Contracting Path

- The **decoder** in U-Net is the **expanding path**.
- Motivation: Upsampling the feature maps from the contracting path, it recovers spatial information and generates the final segmentation map.
- **Components**: Transposed Convolutional Layers + Upsampling layers to upsample the feature maps.
- **Goals**:
  - ➢ Reconstructs the original spatial dimensions via skip connections by integrating the upsampled feature maps with the equivalent maps from the encoder.
  - ➢ Enables the network to recover fine-grained features and properly localize items.



Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 (pp. 234-241). Springer International Publishing.

Skip Connection in U-net



Skip connections (SC) are essential to the UNET design because:

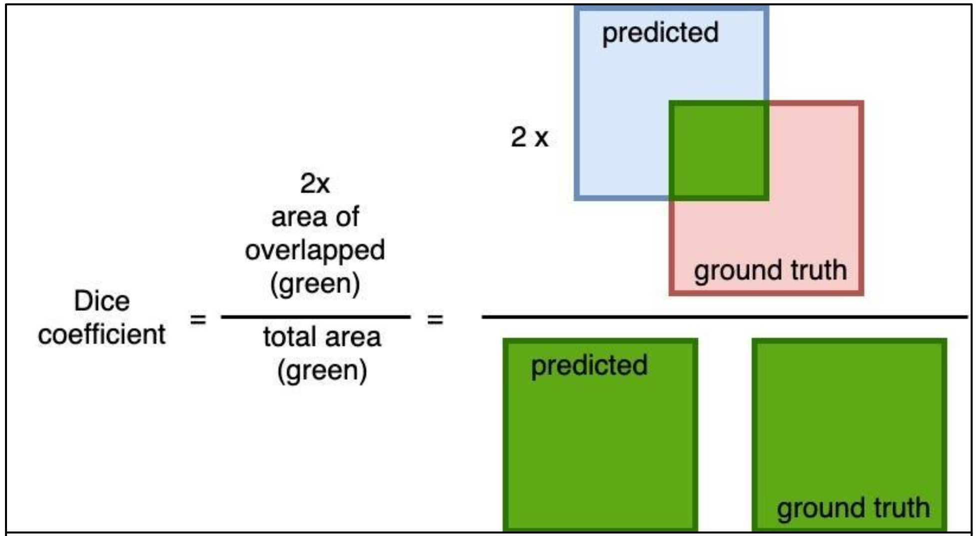
- Allow information to travel between the contracting (encoding) and expanding (decoding) paths.
- Maintaining spatial information and improving segmentation accuracy.

To be specific, the **abilities** of SC are:

- Preserving Spatial Information.
- Multi-scale Information Fusion.
- Combining High-Level Context and Low-Level Details

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 (pp. 234-241). Springer International Publishing.

Loss Function – Dice Loss and CE Loss



$$\mathbf{L}_{dice} = \frac{2 * \sum p_{true} * p_{pred}}{\sum p_{true}^2 + \sum p_{pred}^2 + \epsilon}$$

$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases}$$

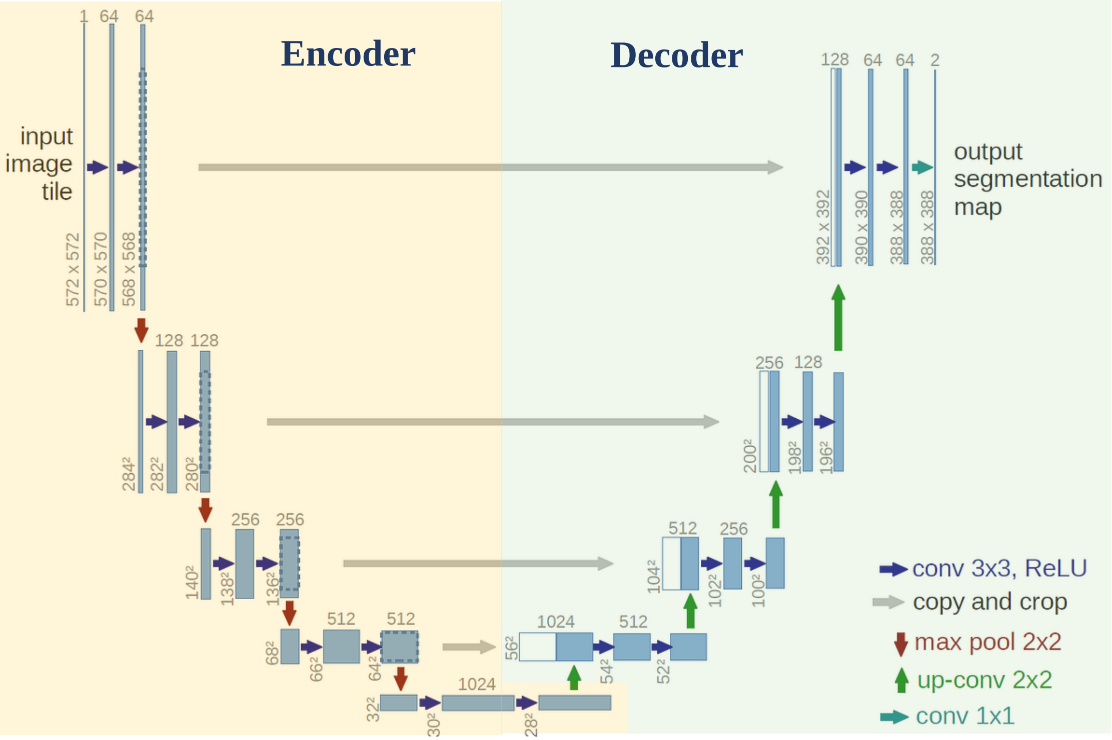
The Dice coefficient is a similarity statistic that calculates the overlap between the anticipated and true segmentation masks. **The Dice coefficient loss**, or **soft Dice loss**, is calculated by subtracting one from the Dice coefficient. When the anticipated and ground truth masks align well, the loss minimizes, resulting in a higher Dice coefficient. **The Dice coefficient loss** is effective for unbalanced datasets in which the background class has many pixels. By penalizing false positives & false negatives, it promotes the network to divide both foreground and background regions accurately.

Cross-Entropy (CE) loss function in image segmentation tasks measures the dissimilarity between the predicted class probabilities and the ground truth labels. Treat each pixel as independent classification tasks in segmentation.
Cross-Entropy loss is computed pixel-wise, effective when:

- Foreground and background classes are balanced.
- Multiple classes are involved in the segmentation task.

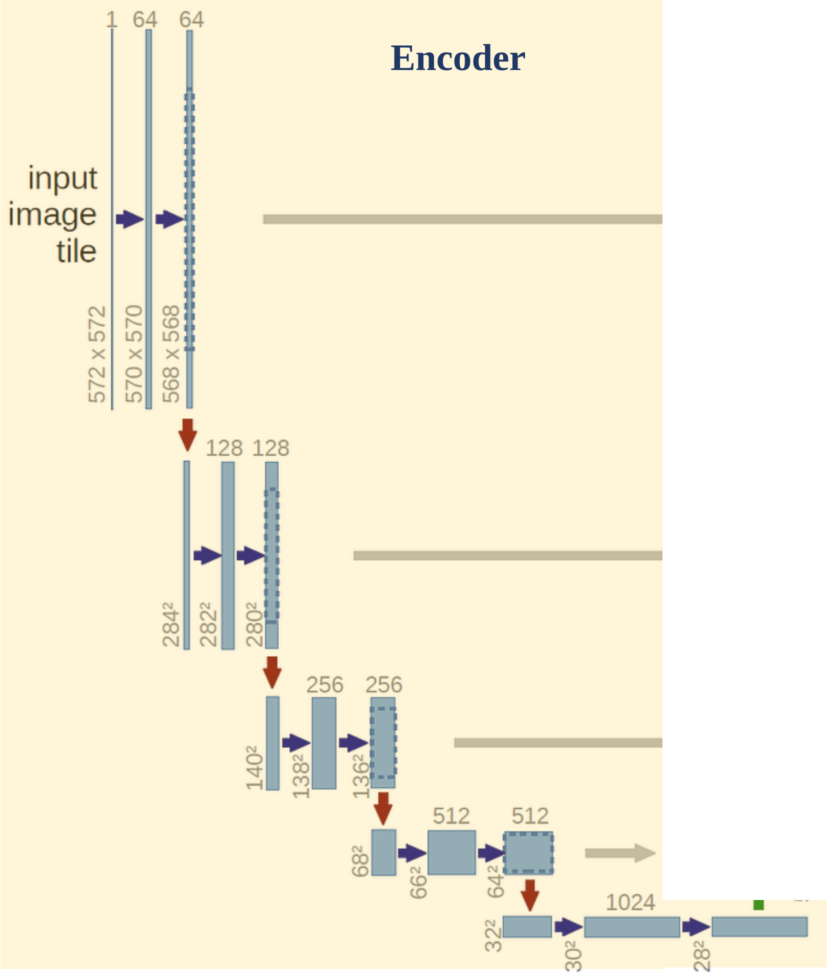
Unet – Demo code

```python
from keras.models import Model, load_model
from keras.layers import Input
from keras.layers.core import Dropout, Lambda
from keras.layers.convolutional import Conv2D, Conv2DTranspose
from keras.layers.pooling import MaxPooling2D
from keras.layers.merge import concatenate
from keras.callbacks import EarlyStopping, ModelCheckpoint
from keras import backend as K

import tensorflow as tf
```




```python
c1 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (s)
c1 = Dropout(0.1) (c1)
c1 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c1)
p1 = MaxPooling2D((2, 2)) (c1)

c2 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (p1)
c2 = Dropout(0.1) (c2)
c2 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c2)
p2 = MaxPooling2D((2, 2)) (c2)

c3 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (p2)
c3 = Dropout(0.2) (c3)
c3 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c3)
p3 = MaxPooling2D((2, 2)) (c3)

c4 = Conv2D(128, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (p3)
c4 = Dropout(0.2) (c4)
c4 = Conv2D(128, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c4)
p4 = MaxPooling2D(pool_size=(2, 2)) (c4)

c5 = Conv2D(256, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (p4)
c5 = Dropout(0.3) (c5)
c5 = Conv2D(256, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c5)

u6 = Conv2DTranspose(128, (2, 2), strides=(2, 2), padding='same') (c5)
u6 = concatenate([u6, c4])
c6 = Conv2D(128, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (u6)
c6 = Dropout(0.2) (c6)
c6 = Conv2D(128, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c6)

u7 = Conv2DTranspose(64, (2, 2), strides=(2, 2), padding='same') (c6)
u7 = concatenate([u7, c3])
c7 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (u7)
c7 = Dropout(0.2) (c7)
c7 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c7)

u8 = Conv2DTranspose(32, (2, 2), strides=(2, 2), padding='same') (c7)
u8 = concatenate([u8, c2])
c8 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (u8)
c8 = Dropout(0.1) (c8)
c8 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c8)

u9 = Conv2DTranspose(16, (2, 2), strides=(2, 2), padding='same') (c8)
u9 = concatenate([u9, c1], axis=3)
c9 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (u9)
c9 = Dropout(0.1) (c9)
c9 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c9)

outputs = Conv2D(1, (1, 1), activation='sigmoid') (c9)
```

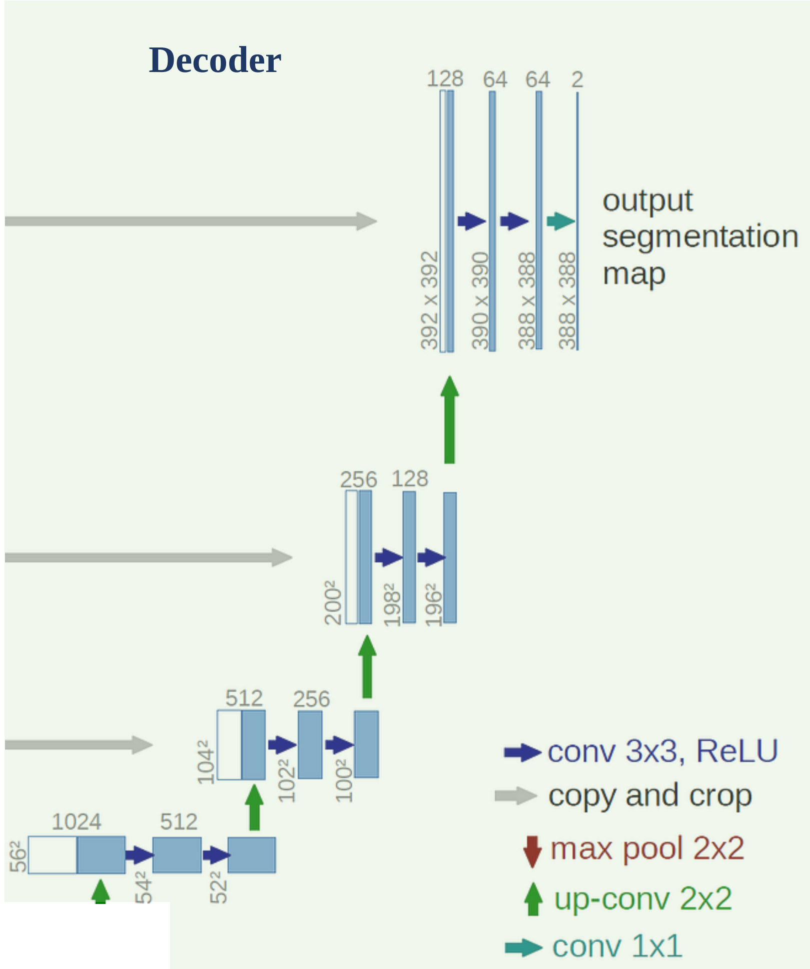
Unet – Demo code




```
c1 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (s)
c1 = Dropout(0.1) (c1)
c1 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c1)
p1 = MaxPooling2D((2, 2)) (c1)

c2 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (p1)
c2 = Dropout(0.1) (c2)
c2 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c2)
p2 = MaxPooling2D((2, 2)) (c2)

c3 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (p2)
c3 = Dropout(0.2) (c3)
c3 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c3)
p3 = MaxPooling2D((2, 2)) (c3)

c4 = Conv2D(128, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (p3)
c4 = Dropout(0.2) (c4)
c4 = Conv2D(128, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c4)
p4 = MaxPooling2D(pool_size=(2, 2)) (c4)

c5 = Conv2D(256, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (p4)
c5 = Dropout(0.3) (c5)
c5 = Conv2D(256, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c5)
```

```
u6 = Conv2DTranspose(128, (2, 2), strides=(2, 2), padding='same') (c5)
u6 = concatenate([u6, c4])
```

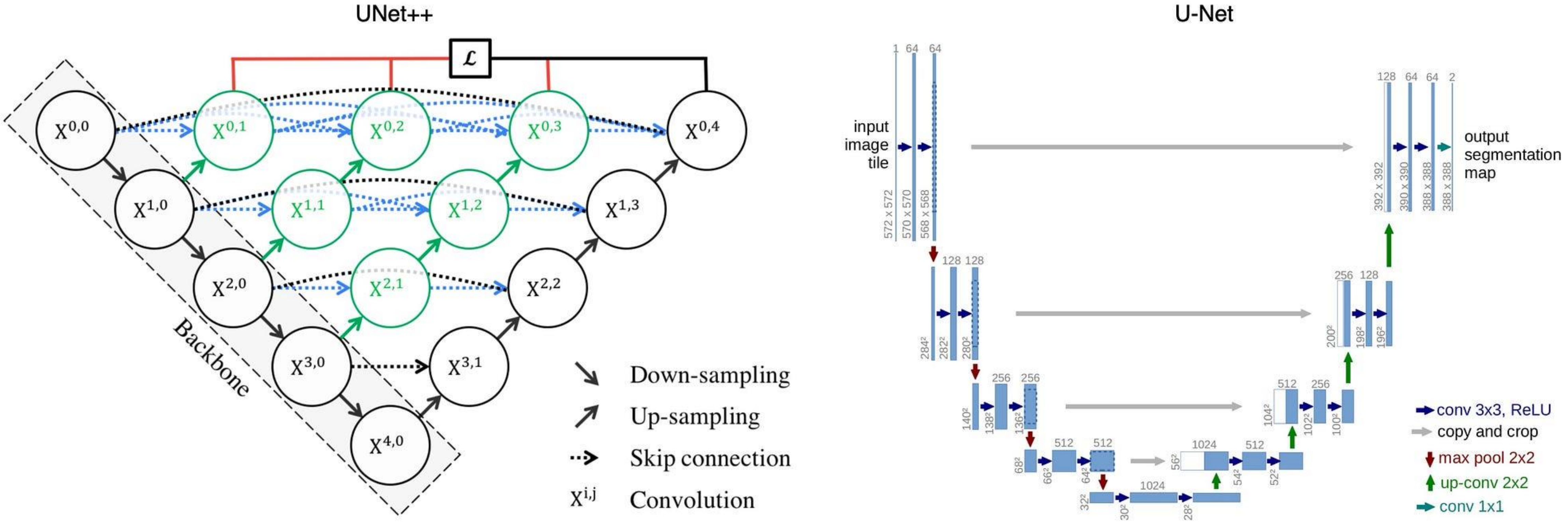
Unet – Demo code




```
u6 = Conv2DTranspose(128, (2, 2), strides=(2, 2), padding='same') (c5)
u6 = concatenate([u6, c4])
c6 = Conv2D(128, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (u6)
c6 = Dropout(0.2) (c6)
c6 = Conv2D(128, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c6)

u7 = Conv2DTranspose(64, (2, 2), strides=(2, 2), padding='same') (c6)
u7 = concatenate([u7, c3])
c7 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (u7)
c7 = Dropout(0.2) (c7)
c7 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c7)

u8 = Conv2DTranspose(32, (2, 2), strides=(2, 2), padding='same') (c7)
u8 = concatenate([u8, c2])
c8 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (u8)
c8 = Dropout(0.1) (c8)
c8 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c8)

u9 = Conv2DTranspose(16, (2, 2), strides=(2, 2), padding='same') (c8)
u9 = concatenate([u9, c1], axis=3)
c9 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (u9)
c9 = Dropout(0.1) (c9)
c9 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (c9)
```

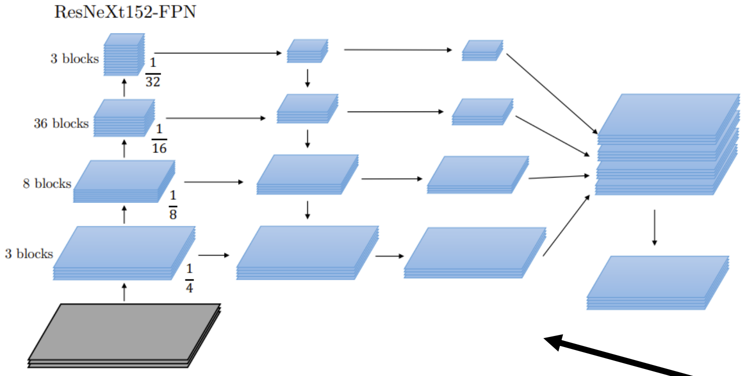
```
outputs = Conv2D(1, (1, 1), activation='sigmoid') (c9)
```

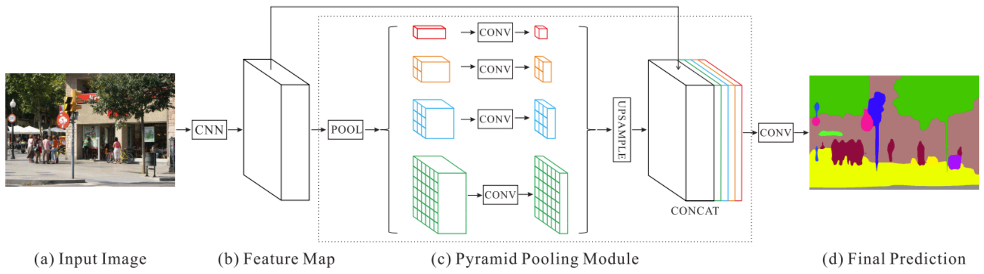
Unet++: A Nested U-Net Architecture



Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., & Liang, J. (2018). Unet++: A nested u-net architecture for medical image segmentation. In Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4 (pp. 3-11). Springer International Publishing.

Other Extensions of U-Net Architecture





- MAnet
- Linknet
- FPN
- PSPNet
- PAN
- DeepLabV3
- DeepLabV3+

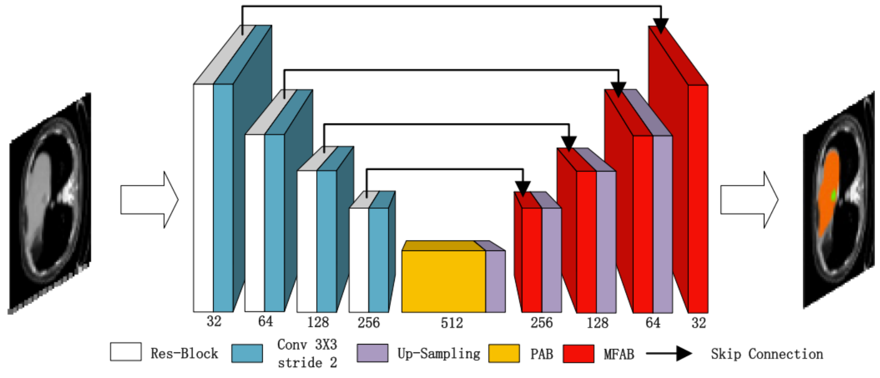
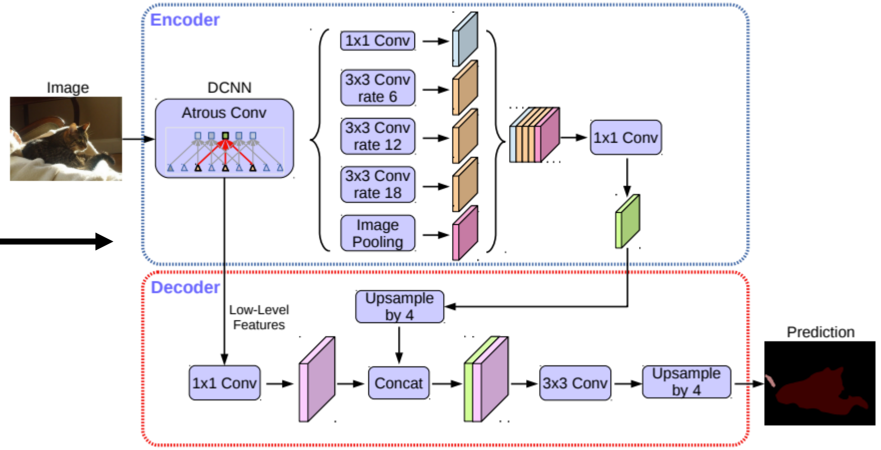
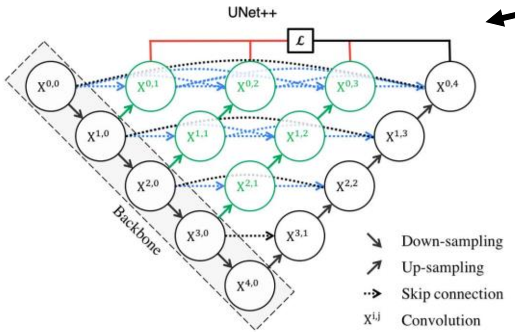
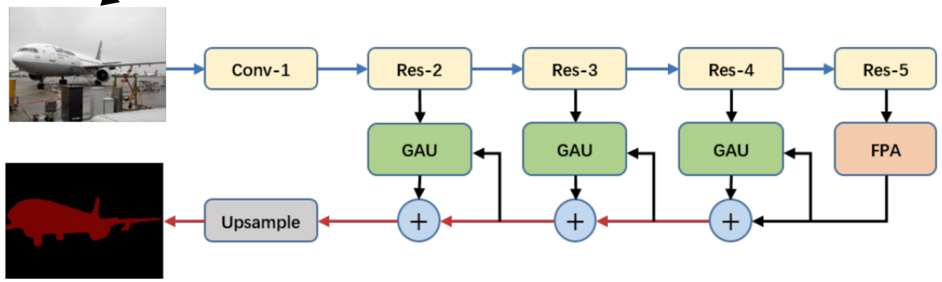


FIGURE 1. The total architecture of MA-Net.







Research Project: Breast Ultrasound Cancer Diagnosis

AI VIET NAM
@aivietnam.edu.vn

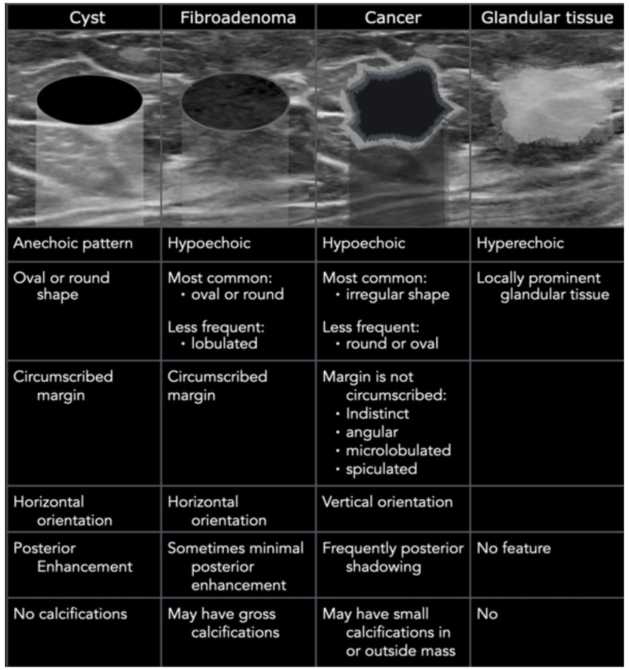
Nguyen Thanh Huy

Breast Ultrasound Segmentation

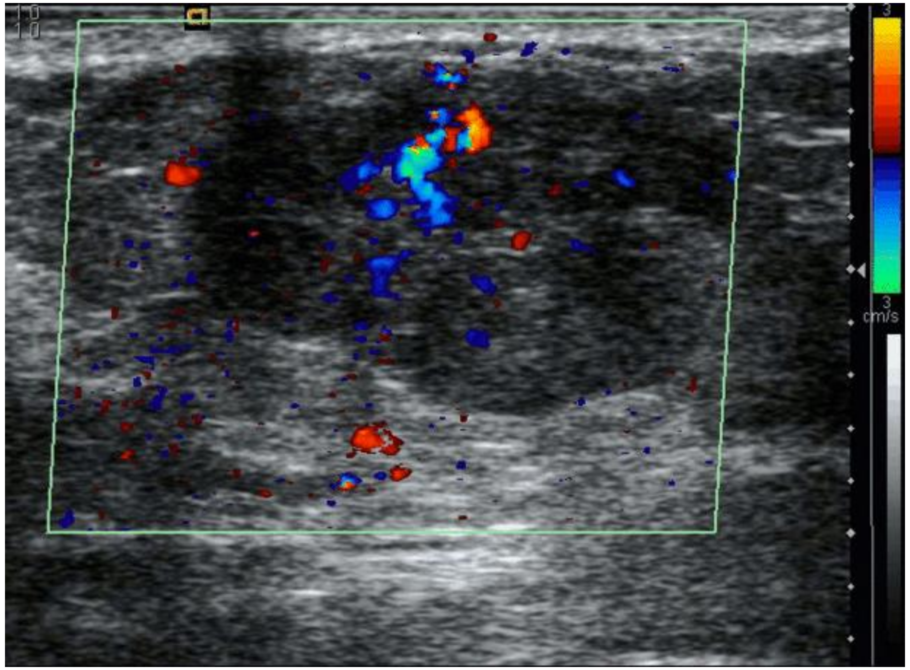
Breast cancer is the leading cause of cancer deaths in women (Highest incidence rate of cancer).

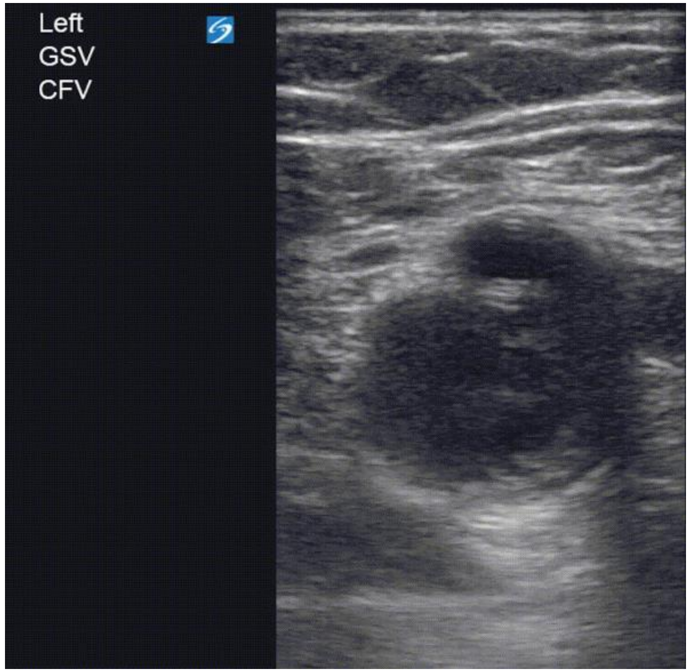
Breast ultrasound imaging is noninvasive, nonradioactive, and cost-effective.

Besides previous studies using x-rays to diagnose cancer mammograms, breast ultrasound diagnosis achieves promising abilities overall.

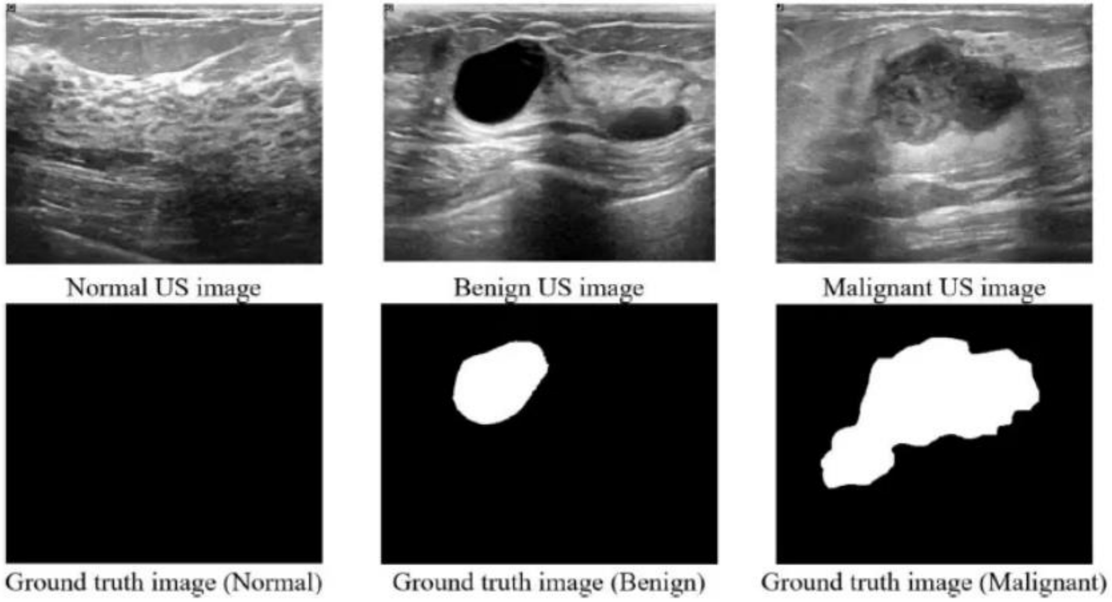


Cyst	Fibroadenoma	Cancer	Glandular tissue
Anechoic pattern	Hypoechoic	Hypoechoic	Hyperechoic
Oval or round shape	Most common: • oval or round Less frequent: • lobulated	Most common: • irregular shape Less frequent: • round or oval	Locally prominent glandular tissue
Circumscribed margin	Circumscribed margin	Margin is not circumscribed: • Indistinct • angular • microlobulated • spiculated	
Horizontal orientation	Horizontal orientation	Vertical orientation	
Posterior Enhancement	Sometimes minimal posterior enhancement	Frequently posterior shadowing	No feature
No calcifications	May have gross calcifications	May have small calcifications in or outside mass	No





BUSI: Dataset of Breast Ultrasound Images



Samples of Ultrasound breast images and Ground Truth Images.

Case	Number of images
Benign	487
Malignant	210
Normal	133
Total	780

The three classes of breast cases and the number of images in each case

The dataset consists of the medical images of **breast cancer using ultrasound** scan, which is categorized into three classes: **normal, benign, and malignant images**.

Breast ultrasound images can produce great results in **classification**, **detection, and segmentation** of breast cancer when combined with machine learning.

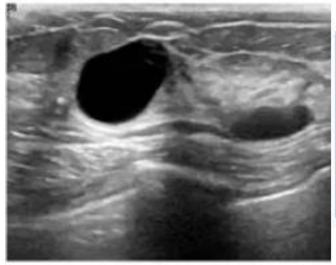
Access: https://scholar.cu.edu.eg/?q=afahmy/pages/dataset

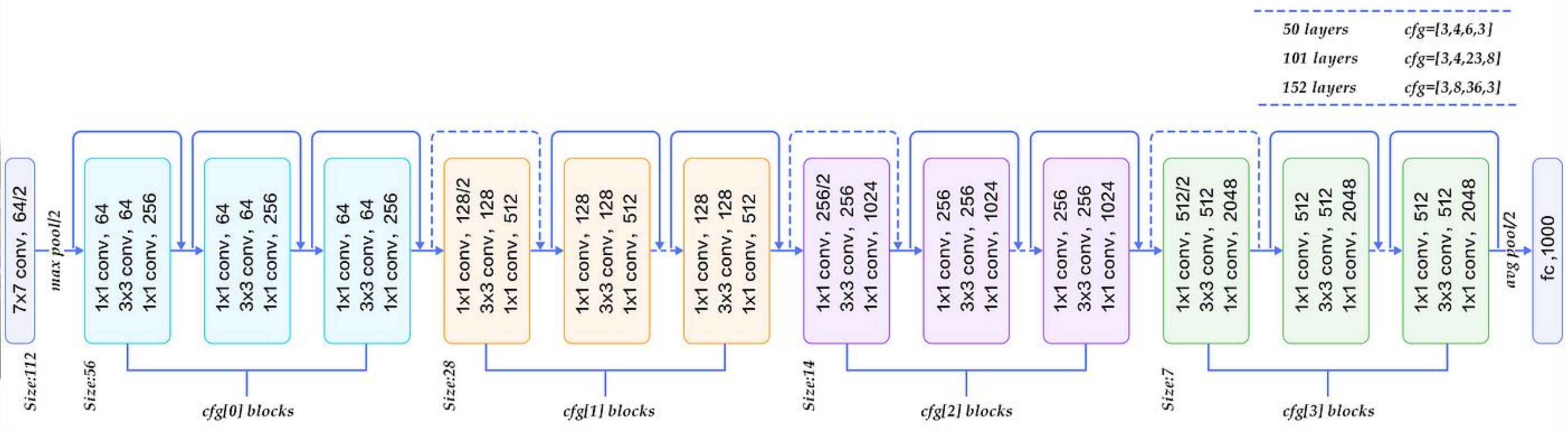
At the beginning, the number of images collected was **1100**. After performing preprocessing to the dataset, the number of images was reduced to **780 images**.

LOGIQ E9 ultrasound system and LOGIQ E9 Agile ultrasound system produce image resolution of **1280×1024**. All images were **cropped**. The average image size of **500×500** pixels.

Single Task Model: ResNet Classification

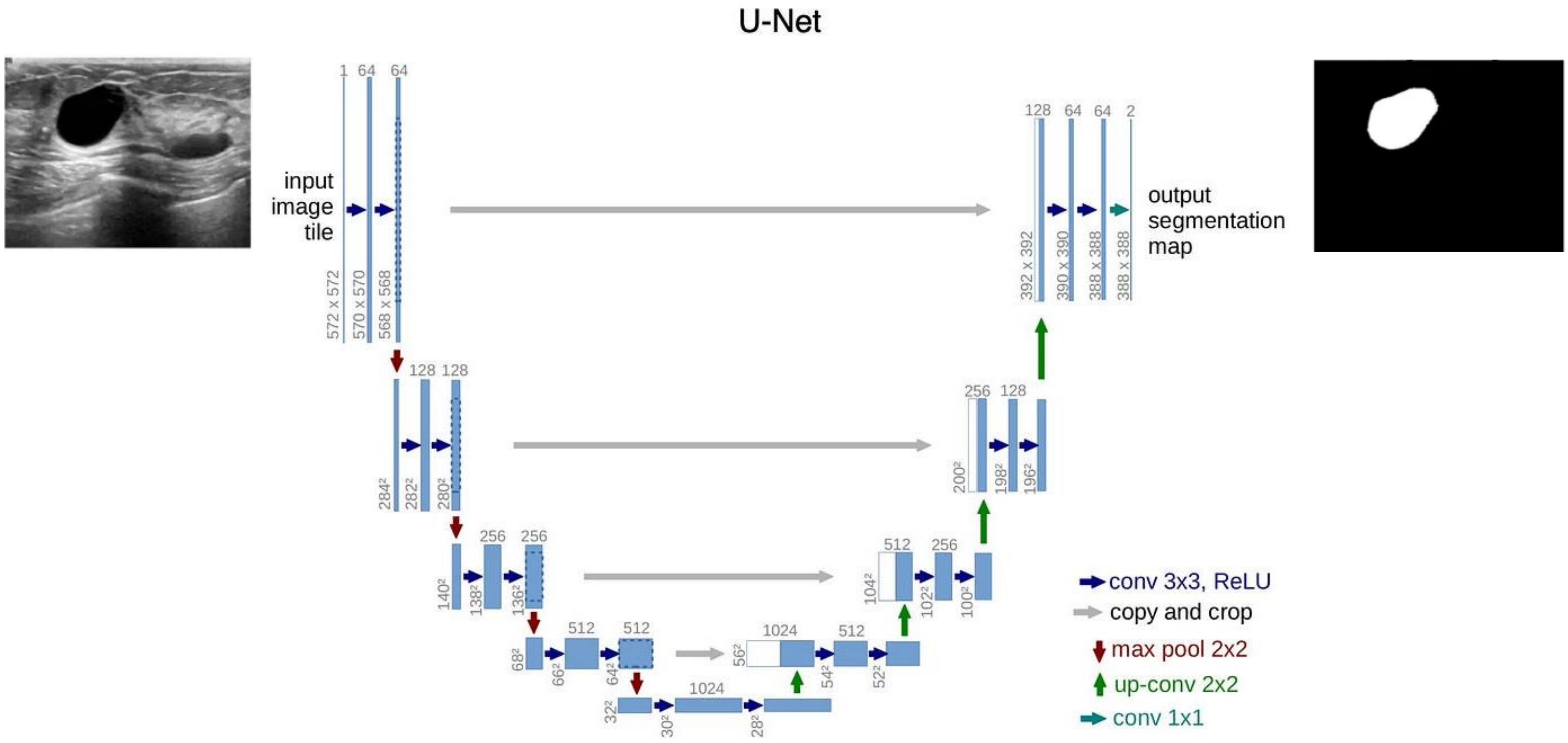
Resnet 50



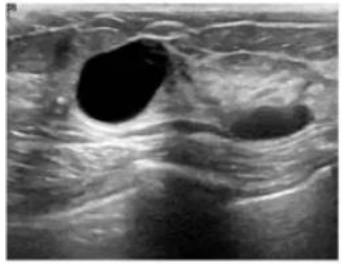


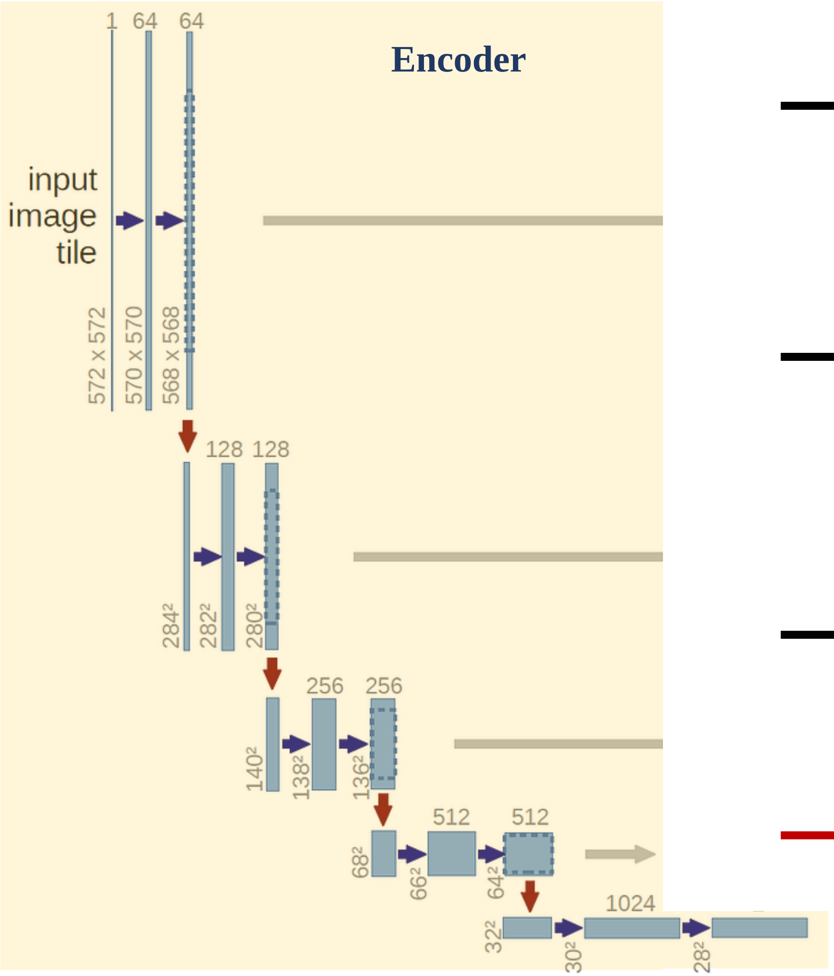
- Normal
- Benign
- Malignant

Single Task Model: U-Net Segmentation



Research Questions & Research Gaps





Encoder
Extract context and compresses the input image by gradually decreasing the spatial dimensions

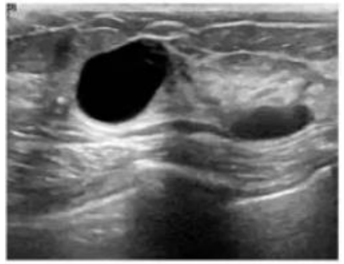
In fact, Encoder is CNN architecture that we already known in the classification task in the previous lessions

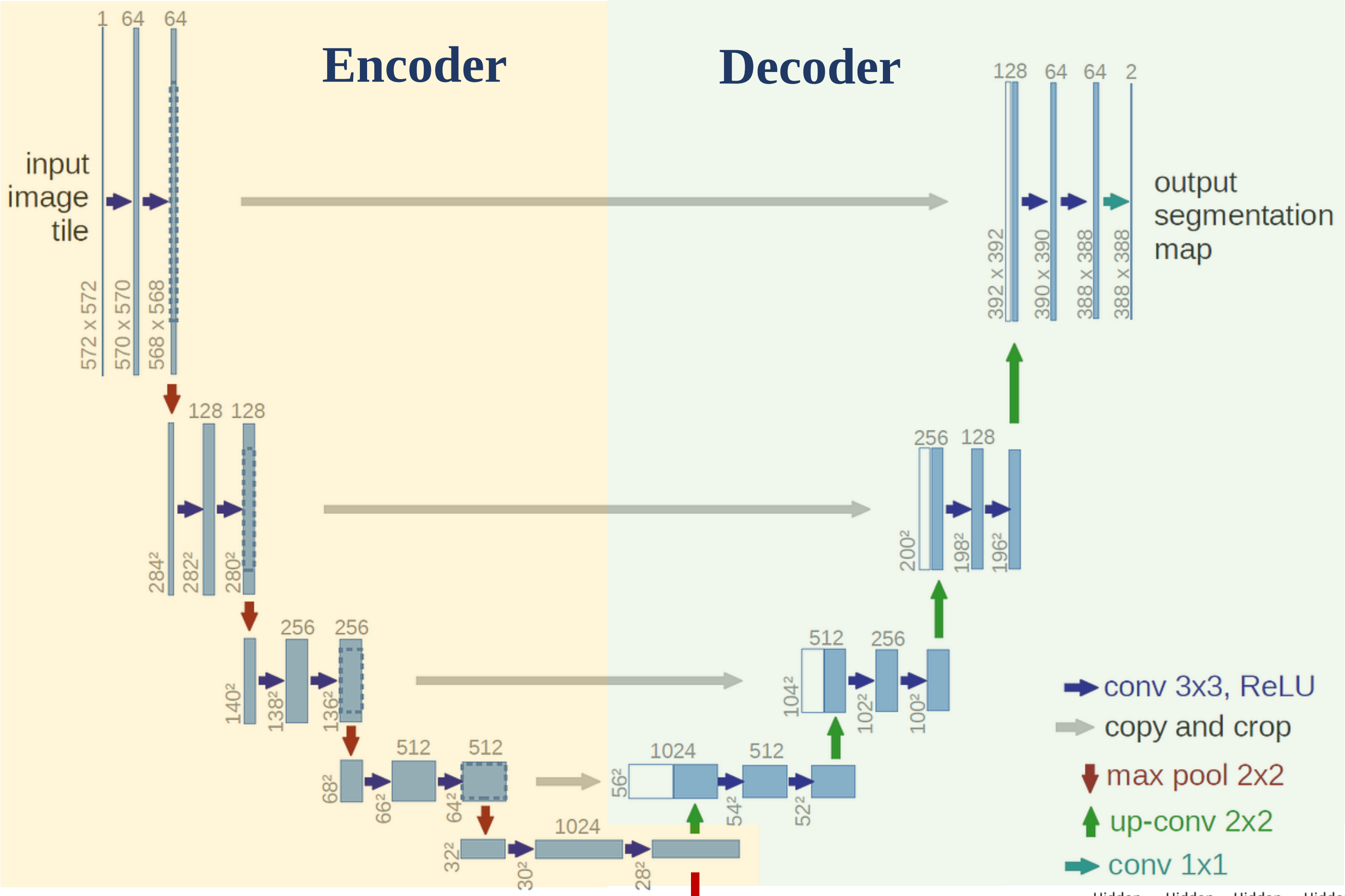
Encoder can be AlexNet, VGG, ResNet, Densenet,…

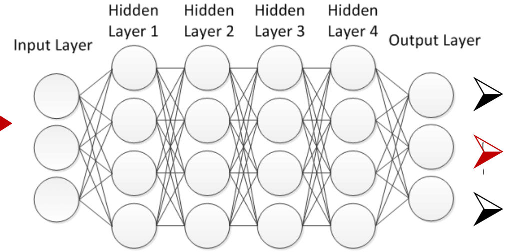
Motivation
Can we redesign U-net model to do both segmentation (seg) task and classification (cls)? (**Encoder** is used for **cls** and **Encoder-Decoder** are used for **seg**)

Yes, We can take the high-level feature from the end of encoder path to **connect with MLP for classification**.

Design Idea for New Proposed Architecture







- Normal
- Benign
- Malignant

Multi-task Model: Seg + Cls

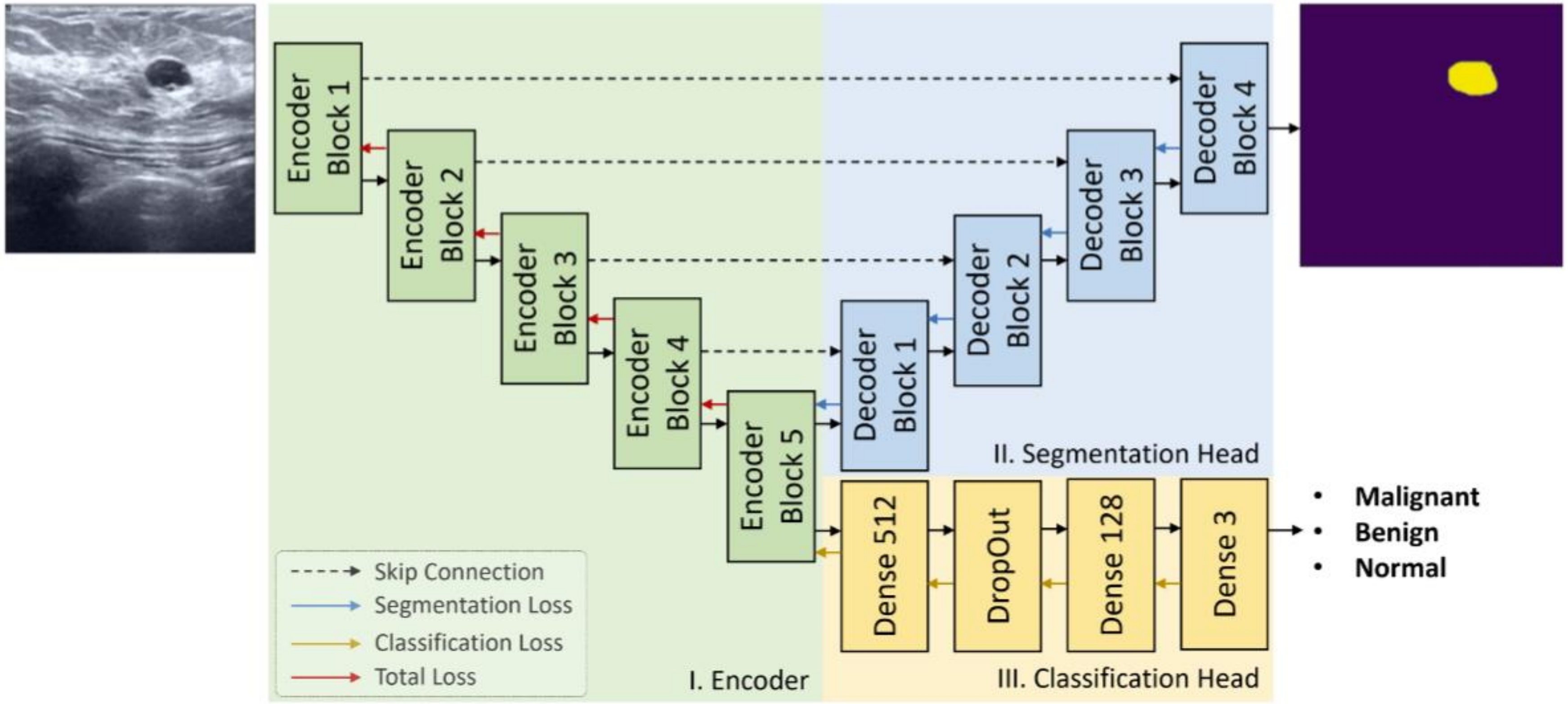


Fig. 1: Visualization of our proposed architecture with multi-task learning approach. The shared encoder (green area) takes weighted sum loss from the segmentation branch (blue area) and classification branch (yellow area).

Research Idea Completion and Research Directions

AI VIET NAM
@aivietnam.edu.vn

Nguyen Thanh Huy

Some suggested research questions

Conference publication

1. There are two tasks, how can we redesign loss functions?
2. If we design combined weighted loss, what is the optimal lambda for best overall or each task performance?
3. Why do we use U-net? What about U-Net++ and others? (Similar to Encoder selection)

Journal publication

4. Single-task learning (STL) or Multi-task learning (MTL) is better?
5. In which encoder and architecture, STL is better than MTL and vice versa?
6. Quantitative results are might be clear, what about qualitative results? (Visualization)
7. What are the advantages and disadvantages of using STL and MTL in visualization (prediction mask)?

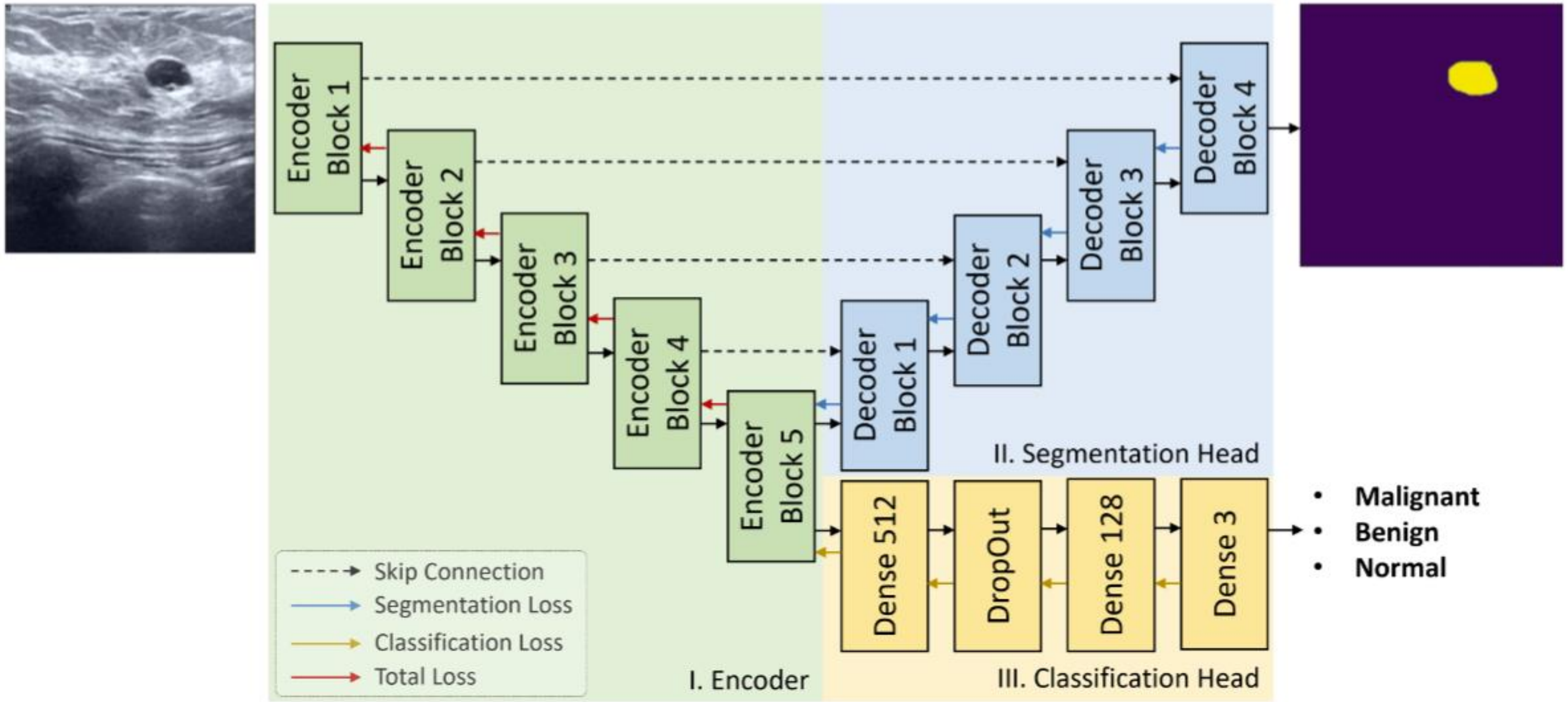
Future Work

(How to overcome the current the disadvantages if you choose to approach MTL?)

8. Are Focal Loss for cls and Dice Loss for seg the best? What are alternative options? (Easy)
9. Transformer shows the better performance compared to CNNs, should we leverage ViT? (Easy)
10. Should we use some techniques of preprocessing, post-processing or data augmentation? (Easy)
11. Can we redesign the architecture, loss or any specific part of framework? (Hard)
12. Can we add module to fill the current limitation of model (e.g. contrastive learning, noisy labeling,…)? (Hard)
13. Can we try some DL-based synthesis approach for resampling? (Hard)

- **For "Hard" contributions, suggestion is make good assumption, have wide knowledge, and understand deeply**.

Question 1: Loss Function?



$$L_{segmentation} = DiceLoss(y, \hat{y}) = \frac{y + \hat{y} + 2y\hat{y}}{y + \hat{y} + \epsilon}$$

$$L_{classification} = FocalLoss(p_t) = -\alpha_t (1 - p_t)^{\gamma} log(p_t)$$

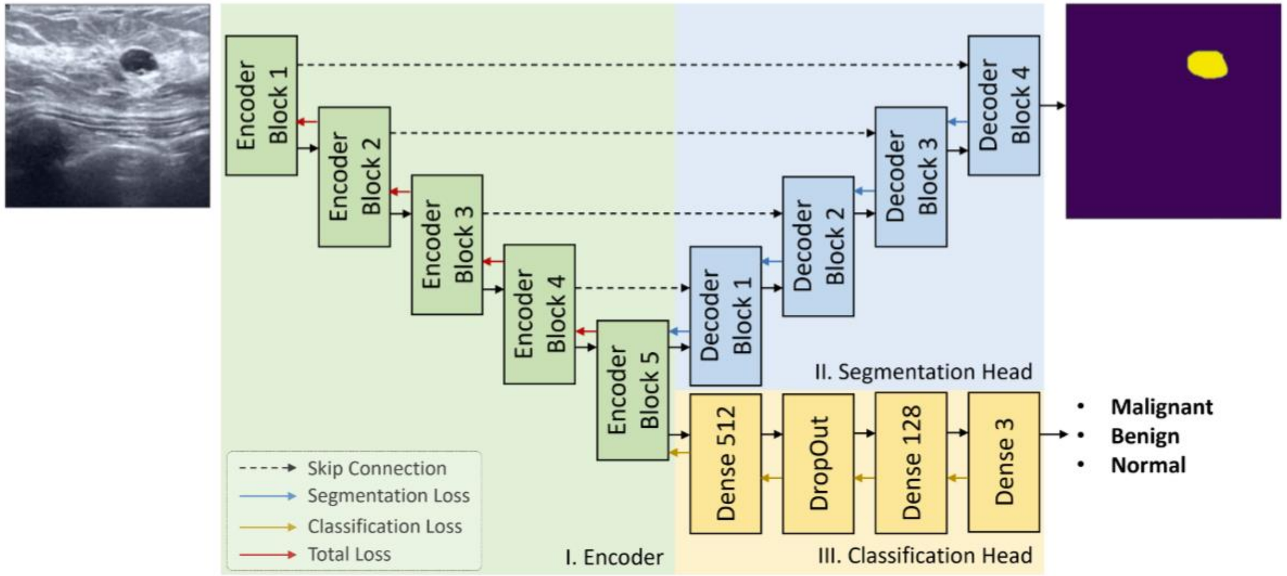
Loss Function

If there are 2 losses, how can the model update loss?

$$L_{total} = \lambda L_{segmentation}(y, \hat{y}) + (1 - \lambda) L_{classification}(p_t)$$

λ denotes the loss weight for each loss element.

Question 2: Best lambda for multi-task?



$$L_{total} = \lambda L_{segmentation}(y, \hat{y}) + (1-\lambda) L_{classification}(p_t)$$

Lambda	Classification				Segmentation			Overall
	Accuracy	F1-score	Precision	Recall	IoU	Dice	F1-score	
0.1	0.713	0.663	0.705	**0.642**	0.604	0.663	0.679	0.656
0.3	0.783	**0.669**	**0.745**	0.635	0.679	0.743	0.759	0.698
0.5	0.828	0.618	0.669	0.586	0.693	0.758	0.774	0.680
0.7	**0.834**	0.658	0.707	0.628	**0.703**	0.766	**0.782**	**0.704**
0.9	0.637	0.594	0.662	0.573	0.69	**0.771**	0.771	0.669

Question 3: Why Unet? Which encoder?

Type	Architecture	Backbone	Classification				Segmentation		
			Acc	F1-Score	Precision	Recall	IoU	Dice	F1-Score
Multitask Model	Unet	Resnet50	0.682	0.648	0.746	0.620	0.682	0.756	0.756
		Resnext50	0.873	0.663	0.694	0.643	0.692	0.758	0.774
		WideResnet50	0.783	0.630	0.695	0.594	0.683	0.763	0.763
		Efficientnet_b4	0.904	0.723	0.754	0.701	0.749	0.817	0.817
	Unet++	Resnet50	0.815	0.637	0.687	0.606	0.662	0.746	0.746
		Resnext50	0.822	0.651	0.704	0.621	0.760	0.771	0.838
		WideResnet50	0.790	0.664	0.738	0.628	0.680	0.761	0.761
		Efficientnet_b4	0.847	0.650	0.694	0.623	0.769	0.795	0.837
	FPN	Resnet50	0.688	0.627	0.679	0.604	0.606	0.601	0.668
		Resnext50	0.745	0.652	0.738	0.621	0.714	0.752	0.794
		WideResnet50	0.758	0.547	0.610	0.518	0.710	0.749	0.790
		Efficientnet_b4	0.796	0.680	0.738	0.649	0.741	0.776	0.817
	DeepLabV3+	Resnet50	0.809	0.676	0.729	0.649	0.715	0.769	0.794
		Resnext50	0.726	0.651	0.746	0.617	0.745	0.781	0.823
		WideResnet50	0.809	0.610	0.669	0.572	0.722	0.797	0.797
		Efficientnet_b4	0.892	0.694	0.726	0.672	0.798	0.813	0.864

Exercises

Nguyen Thanh Huy

Question 1: Loss Function?

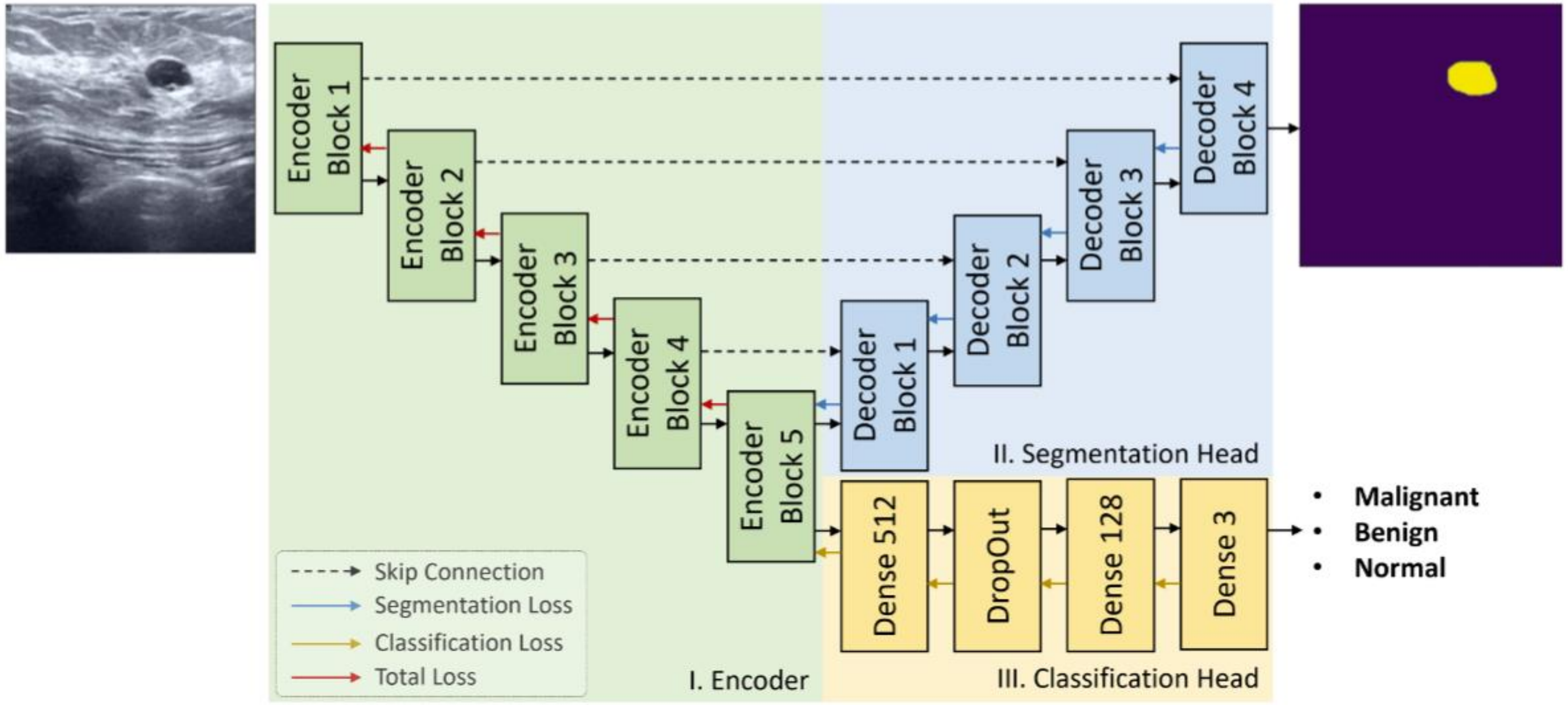


Fig. 1: Visualization of our proposed architecture with multi-task learning approach. The shared encoder (green area) takes weighted sum loss from the segmentation branch (blue area) and classification branch (yellow area).

Kaggle Competition

Competition: *https://www.kaggle.com/t/10e9c6922c354876a0c14dcffa7084b4*

BACH NGO · COMMUNITY PREDICTION COMPETITION · PRIVATE · A MONTH TO GO

Submit Prediction

AIO - Medical Image

Private AIO competition for medical image processing tasks, including object detection and segmentation.

Overview Data Code Models Discussion Leaderboard Rules

Overview

Breast cancer is the leading cause of cancer deaths in women. It currently has the highest incidence rate of cancer among women in the United States; 31% of all newly diagnosed cases of cancer in 2022 were found to be related to it. Because of its high incidence rate, early detection of breast cancer is crucial to lowering death rates and increasing available treatment options.

This is a competition organized by the AIO TA team to effectively tackle this problem

Start
7 days ago

Close
a month to go

Competition Host
Bach Ngo

Prizes & Awards
Kudos
Does not award Points or Medals

Participation
0 Competitors
0 Teams
0 Entries

Table of Contents
Overview

Kaggle Competition

Baseline: *https://www.kaggle.com/code/bachngoh/aio-icisn-homework-in-class/notebook*

[AIO] ICISN_ Homework_In_Class

0 Edit

Notebook Input Output Logs Comments (0) Settings

```
#Solver
CLASSES = {0: "Benign", 1: "Malignant", 2: "Normal"}
INPUT_SIZE = (448,448)
BATCH_SIZE = # Điền batch size vào đây
BASE_LR = # Điền learning rate phù hợp vào đây
MAX_EPOCHS = # Số epoch train
SAVE_INTERVAL = 10
PATIENCE = 300
N_SPLITS = # điền số K folds vào đây

#Model
ARCH = # chọn giữa ['unet', 'unetpp', , 'fpn', 'deeplabv3plus']
ENCODER_NAME = # chọn giữa các kiến trúc ['resnet50', 'resnext50_32x4d', 'tu-wide_resnet50_2', 'efficientnet-b4']
IN_CHANNELS = 3
SEG_NUM_CLASSES = 2
CLA_NUM_CLASSES = 3
OUTPUT_ACTIVATION = None #None for logits

#Loss coefficient weight
ALPHA = # chọn số alpha hợp lý

#Path
OUTPUT_DIR = # điền vào đây
DATASET_DIR = # điền vào đây
CHECKPOINT = None
```

Table of Contents

Inference Notebook

Inference: *https://www.kaggle.com/code/bachngoh/aio-medical-image-analysis-infer*

[AIO] Medical Image Analysis - Infer

0 Edit

Notebook Input Output Logs Comments (0) Settings

```
NUM_WORKERS = 0
DEVICE = torch.device("cuda" if torch.cuda.is_available() else "cpu")

#Solver
CLASSES = {0: "benign", 1: "malignant", 2: "normal"}
INPUT_SIZE = (448,448)
BATCH_SIZE = 8
BASE_LR = 0.01
MAX_EPOCHS = 3
SAVE_INTERVAL = 10
PATIENCE = 300
N_FOLDS = 3


#Model
ARCH = "deeplabv3plus" # chọn giữa ['unet', 'unetpp', , 'fpn', 'deeplabv3plus']
ENCODER_NAME = "efficientnet-b4" # chọn giữa các kiến trúc ['resnet50', 'resnext50_32x4d', 'tu-wide_resnet50_2', 'efficientnet-b4']
IN_CHANNELS = 3
SEG_NUM_CLASSES = 2
CLA_NUM_CLASSES = 3
OUTPUT_ACTIVATION = None #None for logits

#Loss coefficient weight
ALPHA = 0.7
```

Table of Contents

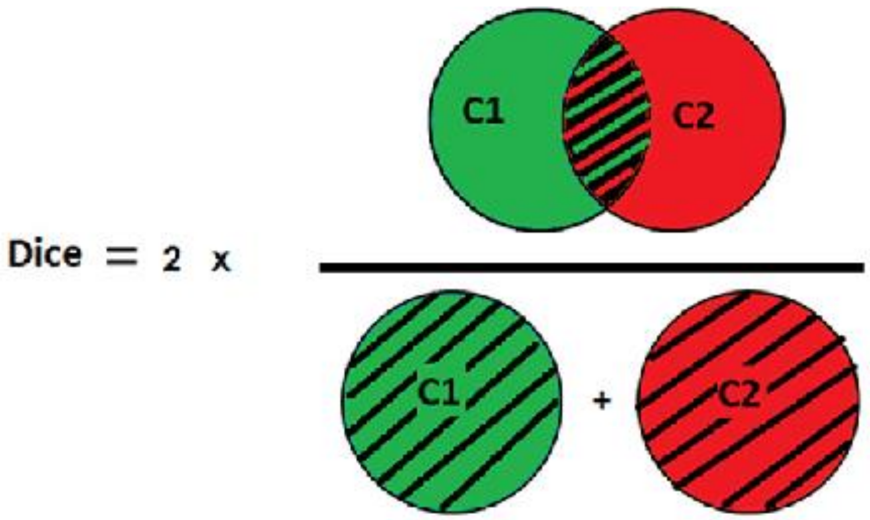
Metrics

ACTUAL VALUES	POSITIVE	NEGATIVE
POSITIVE	TP	FN
NEGATIVE	FP	TN

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$F1\,Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$



Rewards

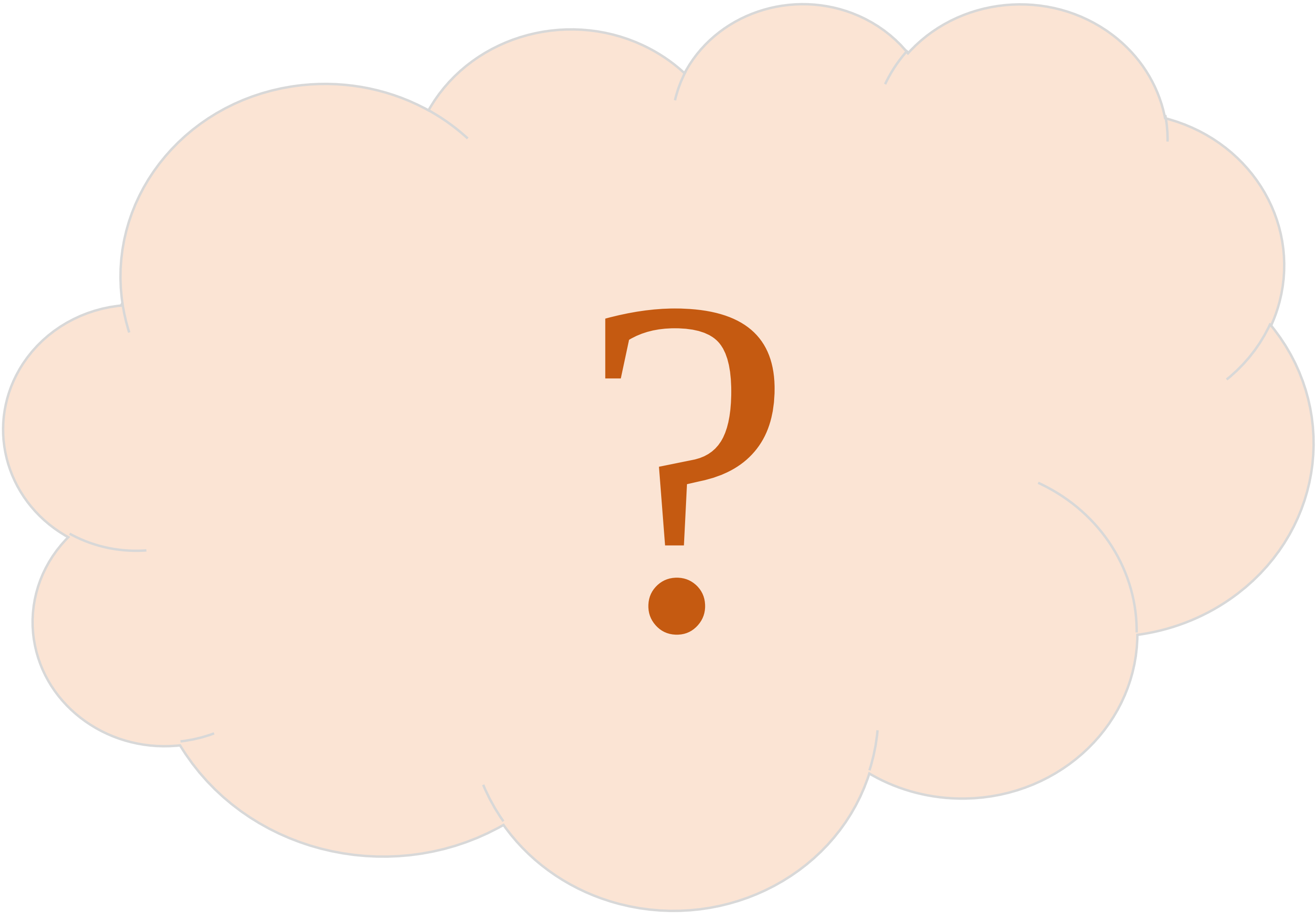
Publications

- ✓ Be co-authored of our on-going paper.
- ✓ If novelty is unique, the work will be supported towards separate publication.

Opportunities

- ✓ Be offered many abroad research opportunities. (Including RA job, or Exchange and PhD offers)
- ✓ Invited to join AIMI.

Questions




Thank you!