Ưu Tiên 1:

* Tác giả thực hiện nhiều thao tác trước khi đưa vào xử lý hình ảnh:
* Resize ảnh về kích thước 96x96 để thống nhất dữ liệu đầu vào:

img = np.uint8(resize(imread(fn), (img\_h, img\_w), order=0, mode='constant', preserve\_range=True))

* Chuyển ảnh grayscale và thêm kênh thành dạng (96,96,1).

img = np.expand\_dims(img, axis=2)

* One-hot encoding ảnh nhãn thành 7 lớp để mô hình SegNet có thể huấn luyện.

def gray2classes(label\_img, img\_h=96, img\_w=96, n\_classes=7):

label\_img\_oh = np.zeros((img\_h, img\_w, n\_classes), dtype=np.uint8)

def get\_oh(label\_img, class\_color):

oh\_tmp = np.zeros((img\_h, img\_w), dtype=np.uint8)

oh\_tmp[np.where(label\_img == class\_color)] = 1

return oh\_tmp

# Định nghĩa màu pixel cho từng lớp

class\_map = {

255: 0, # Lòng bàn tay

43: 1, # Ngón 1

85: 2, # Ngón 2

128: 3, # Ngón 3

170: 4, # Ngón 4

212: 5, # Ngón 5

0: 6 # Nền (background)

}

for class\_color, class\_index in class\_map.items():

label\_img\_oh[:, :, class\_index] = get\_oh(label\_img, class\_color)

return label\_img\_oh

* Reshape dữ liệu nhãn từ (96,96,7) thành (số ảnh, 9216, 7) để phù hợp với softmax().

data\_y = data\_y.reshape((int(len(loi)), img\_h \* img\_w, n\_classes))

Hiển thị ảnh gốc và ảnh nhãn phân đoạn để kiểm tra dữ liệu.

plt.imshow(x\_test[1000, :, :, 0], cmap='gray')

plt.show()

plt.figure(figsize=(18, 5))

for fin\_ind in range(1, 8): # 7 lớp

plt.subplot(1, 7, fin\_ind)

plt.imshow(y\_test[1000, :, fin\_ind - 1].reshape(img\_h, img\_w), cmap='gray')

plt.show()

* Mục tiêu: Chuẩn bị dữ liệu phù hợp để SegNet phân đoạn ảnh X-quang chính xác!
* Quá trình Train mô hình:
* Ảnh đầu vào (x\_train): Ảnh grayscale của bàn tay, kích thước **96×96×1**.
* Nhãn (y\_train): One-hot encoding phân đoạn các bộ phận của bàn tay (ngón tay, lòng bàn tay, nền).
* Sử dụng **SegNet**, một dạng U-Net đơn giản hóa, có kiến trúc Encoder-Decoder.

lr = 0.0001

batch\_size = 8

nb\_epoch = 15

optimizer = Adam(lr=lr)

autoencoder.compile(loss="categorical\_crossentropy", optimizer=optimizer, metrics=['accuracy'])

autoencoder.fit(x\_train, y\_train,

validation\_data=(x\_test, y\_test),

batch\_size=batch\_size, epochs=nb\_epoch,

verbose=1)

* **Hàm mất mát (Loss Function)**: categorical\_crossentropy phù hợp với phân loại đa lớp.
* **Bộ tối ưu hóa (Optimizer)**: Adam với learning rate 0.0001.
* **Train trong 15 epochs** với batch size **8**.
* **Validation** bằng tập x\_test, y\_test.
* Ưu Tiên 2: Chưa Hoàn thành được vì chưa train file h5 để tiến hành train dữ liệu cho Seg-unet.
* Ưu tiên 3:
* Tổng quan về bộ dữ liệu BTXRD
* **Số lượng ảnh:** 3,746 ảnh X-quang
* 1,879 ảnh bình thường (không có khối u)
* 1,867 ảnh có khối u
* Mỗi ảnh có:
* Thông tin lâm sàng
* Nhãn toàn cục (**classification**)
* **Bounding box** cho từng khối u (**localization**)
* **Mask phân vùng khối u** (**segmentation**)
* **Cách chia tập dữ liệu**
* Theo bài báo, bộ dữ liệu **BTXRD** được chia như sau​:
* **Training set:** 70% dữ liệu (~2,622 ảnh)
* **Validation set:** 15% dữ liệu (~562 ảnh)
* **Test set:** 15% dữ liệu (~562 ảnh)
* So sánh với phương pháp chia dữ liệu phổ biến

A screenshot of a phone

AI-generated content may be incorrect.

* **Nhận xét:**
* **BTXRD sử dụng tỷ lệ 70-15-15%**, giúp đảm bảo dữ liệu test không quá nhỏ nhưng vẫn có đủ ảnh để train.
* **So với cách chia 80-10-10% phổ biến**, phương pháp của bài báo có tập **validation lớn hơn**, giúp điều chỉnh mô hình tốt hơn.
* **Không dùng K-fold Cross Validation**, có thể do kích thước dataset lớn.

Khảo sát;

1. **Deep Learning Image Segmentation Approaches for Malignant Bone Lesions (2024)**

* **Hướng tiếp cận**: Tổng hợp và phân tích các phương pháp deep learning cho phân đoạn khối u xương ác tính.
* **Bài toán**: Sử dụng CNN, đặc biệt là **U-Net**, để phân đoạn khối u trong ảnh X-quang, CT, MRI.
* **Kết quả đạt được**:
  + **IoU trung bình**: Cải thiện đáng kể so với phương pháp cổ điển.
  + **CNN (U-Net) vượt trội hơn so với các phương pháp truyền thống như Watershed và Active Contour**.

1. **Deep Learning for Bone Tumor Detection and Segmentation: 2D vs 3D (2024)**

* **Hướng tiếp cận**: So sánh hiệu suất giữa phương pháp **2D** và **3D Deep Learning** trong MRI khối u xương.
* **Bài toán**: Xác định xem **3D segmentation có vượt trội hơn 2D slice-based segmentation không**.
* **Kết quả đạt được**:
  + **3D Segmentation giúp cải thiện độ chính xác** nhưng yêu cầu tài nguyên tính toán cao hơn.
  + **Hybrid 2D-3D kết hợp giúp cân bằng giữa hiệu suất và tốc độ xử lý**.

1. **Bone Tumor Prediction using Machine Learning and Deep Learning (2025)**

* **Hướng tiếp cận**: Phát hiện và dự đoán sự phát triển của khối u xương bằng AI.
* **Bài toán**: Xây dựng mô hình AI để phân loại và dự đoán tiến triển khối u từ ảnh X-quang.
* **Kết quả đạt được**:
  + **Độ chính xác Classification đạt trên 95% khi kết hợp CNN và Random Forest**.
  + **Tích hợp AI vào hệ thống chẩn đoán sớm giúp giảm sai sót trong phân loại khối u**.

1. **Emerging Applications of Deep Learning in Bone Tumors (2024)**

* **Hướng tiếp cận**: Ứng dụng AI trong chẩn đoán và phân đoạn khối u xương.
* **Bài toán**: Kết hợp **Deep Learning và Radiomics** để nâng cao hiệu quả chẩn đoán.
* **Kết quả đạt được**:
  + **AI giúp giảm sai sót trong chẩn đoán tới 30% so với phương pháp truyền thống**.
  + **Deep Learning có thể phân loại chính xác nhiều loại khối u, đặc biệt là Osteosarcoma**.