Liver Segmentation from CT images using Deep Learning

Soubith Babu Kanatan
20BAI1177
SCOPE
VIT Chennai

Girisankar G
20BAI1259
SCOPE
VIT Chennai

Dr. Geetha S
Professor and Associate Dean (Research)
SCOPE
VIT Chennai

***Abstract* —** Liver cancer is one of the leading causes of cancer-related deaths worldwide. Early detection and treatment are crucial for improving patient outcomes. Accurate segmentation of liver tumors in medical images, such as CT scans and MRI images, is essential for this process. UNet, a convolutional neural network (CNN) architecture, has emerged as a powerful tool for image segmentation, demonstrating remarkable performance in various medical imaging applications. This project delves into the application of UNet for liver tumor segmentation, exploring its architecture, training process.

UNet comprises a contracting path and an expansive path, each composed of convolutional and pooling or upsampling layers, respectively. Skip connections bridge these paths, enabling the network to capture contextual information from both high-resolution and low-resolution feature maps. This combination of skip connections and multi-resolution feature extraction enables UNet to excel in precise localization of features, particularly beneficial for segmenting intricate structures like liver tumors.

Keywords: Liver tumors, CT, UNet, CNN, contracting path, expansive path, skip connections

I. INTRODUCTION

Hepatocellular carcinoma (HCC), another name for liver cancer, is a kind of cancer that starts in the liver, which is the biggest solid organ in the human body. The liver is essential for several body processes, such as bile generation, protein synthesis, and detoxification. Any area of the liver can develop liver cancer, which can range in kind from benign (non-cancerous) to malignant (cancerous). Diagnosing liver cancer typically involves a combination of imaging tests, blood tests, and liver biopsies. Over 905,700 people were diagnosed with and 830,200 people died from liver cancer globally in 2020. Liver cancer was among the top three causes of cancer death in 46 countries. With current trends in these numbers and causes, according to different studes, the number of new cases and deaths from liver cancer could rise by >55% by 2040.

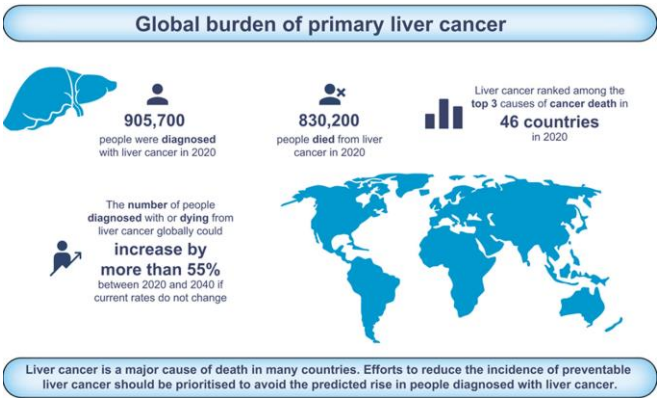
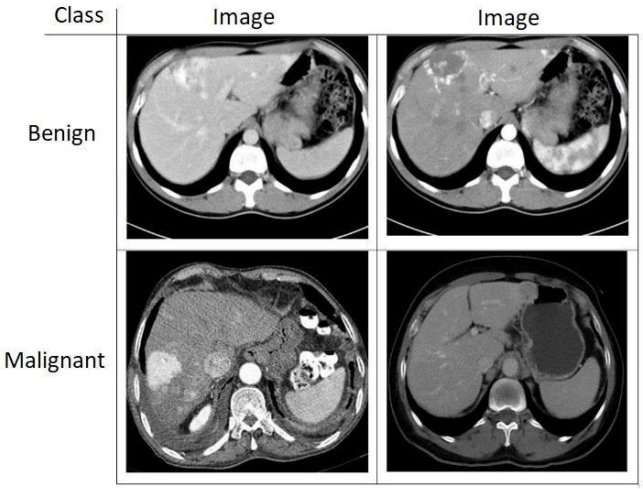


Fig. Stats on global Liver Cancer
https://www.journal-of-hepatology.eu/article/S0168-8278%2822%2903022-7/fulltext

Liver tumor segmentation is the process of identifying the boundaries of a liver tumor in a CT image. CT images can be used to detect liver tumors and to track their growth over time. Deep learning has been shown to be very effective for liver tumor segmentation, and it is now the most widely used method for this task.



II. LITERATURE SURVEY

In, [1], Network is composed of one shared encoder and two inference branches used for segmentation and classification and takes the concatenation of an input image and two Euclidean distance maps. [2] showed a DCNN model design which belongs to the category of "fully convolutional neural networks" (FCNs). It allows complete segmentation of an entire (2D) image in a single pass instead of classifying the center pixel of a small image patch each time. Cascaded Fully Convolutional Neural Networks (CNNs) which involves cascaded 2D CNNs to perform coarse segmentation of the liver. 3D Conditional Random Fields (CRFs) are used to incorporate contextual information and refine the segmentation results in [3]. [4] uses UNet architecture, a groundbreaking deep learning model designed specifically for biomedical image segmentation tasks. The approach in [5] used combines voxel classification to differentiate between tumor and non-tumor regions with affinity constraint propagation, which considers spatial relationships among neighboring voxels to refine tumor boundaries. The methodology in [6] combines two key components: a Hidden Markov Measure Field (HMMF) model and non-parametric distribution estimation. The method in [7] utilizes a mathematical framework based on Discriminant Grassmannian Manifolds, a technique that allows for the representation and analysis of data in a high-dimensional space. This approach aims to capture the unique characteristics of metastatic liver tumors, which can vary significantly in shape and appearance. In [8], a statistical shape model is constructed based on PCA and the input image is smoothed using curvature anisotropic diffusion filtering. In the second step, the mean shape model is moved by using thresholding and Euclidean distance transformation to obtain a coarse position in a test image, and then the initial mesh is locally and iteratively deformed to the coarse boundary. [9] explains a novel three-dimensional (3D) affine invariant shape parameterization is employed to compare local shape across organs. By generating a regular sampling of the organ's surface, this parameterization can be effectively used to compare features of a set of closed 3D surfaces point-to-point, while avoiding common problems with the parameterization of concave surfaces. In [10], the authors describe a fully automatic method for liver tumor segmentation, which employs cascaded CNNs to provide a liver and tumor segmentation. They additionally filter the tumor candidates by means of shape and image-based features computed for each candidate. The authors in [11] evaluate a semi-automatic liver segmentation method based on a level-set approach and a dynamically adapted speed function. The approach also relies on a-priori anatomic information to reduce leakage at the liver-rib interface. The numerical algorithms have been integrated into a complete system that permits loading DICOM images, segmenting these images, and visualizing the liver surface. In [12], a novel 3D deeply supervised network (3D DSN) to address this challenging task. The proposed 3D DSN takes advantage of a fully convolutional architecture which performs efficient end-to-end learning and inference. More importantly, we introduce a deep supervision mechanism during the learning process to combat potential optimization difficulties. A conditional random field model is further employed to obtain refined segmentation. In [13], a framework for automated liver segmentation via a level set formulation is presented. A sparse representation of both global (region-based) and local (voxel-wise) image information is embedded in a level set formulation to innovate a new cost function. [14] proposes a novel hybrid densely connected UNet (H-DenseUNet), which consists of a 2D DenseUNet for efficiently extracting intra-slice features and a 3D counterpart for hierarchically aggregating volumetric contexts under the spirit of the auto-context algorithm for liver and tumor segmentation. In [15], an automatic segmentation framework based on 3D convolutional neural

network (CNN) and globally optimized surface evolution is made. A two-stage framework for 2D liver and tumor segmentation is proposed in [16]. The first stage is a coarse liver segmentation network, the second stage is an edge enhanced network (E 2Net) for more accurate liver and tumor segmentation. E 2Net explicitly models complementary objects (liver and tumor) and their edge information within the network to preserve the organ and lesion boundaries. Edge prediction module in E2Net and design an edge distance map between liver and tumor boundaries, which is used as an extra supervision signal to train the edge enhanced network. A fully automatic approach for liver segmentation is introduced in [17]. The approach consists of three main stages; pre-processing, segmentation and post processing. Watershed segmentation algorithm is used in the main processing stage to detect the borders and edges accurately between the liver regions and the background. In [18], the detection algorithm uses an object-based image analysis approach, allowing for effectively integrating domain knowledge and reasoning processes into the detection logic. [19] proposes a novel context aware PolyUNet for accurate liver and lesion segmentation. It jointly explores structural diversity and consecutive t adjacent slices to enrich feature expressive power and spatial contextual information while avoiding the overload of GPU memory consumption. In [20] a novel deep learning network (cascade U-ResNets) to produce liver and lesion segmentation simultaneously. [21] describes a semi-automated segmentation method for the liver and evaluate its performances on CT-Scan and MRI images. First, an approximate 3D model of the liver is initialized from a few user-generated contours to globally outline the liver shape. The model is then automatically deformed by a Laplacian mesh optimization scheme until it precisely delineates the patient's liver. A correction tool was implemented to allow the user to improve the segmentation. In [22], an efficient algorithm is proposed to enhance the edge information of the CT images prior to segmentation, by means of multislice image fusion and anisotropic diffusion filtering in non-subsampled contourlet (NSCT) domain. In [23], a deep learning-based segmentation algorithm is employed for liver and tumor segmentation from abdominal CT scan images. Three separate UNet models, one for liver segmentation and the others two for tumor segmentation from the segmented liver and directly from the abdominal CT scan image were used deep learning-based segmentation algorithm is employed for liver and tumor segmentation from abdominal CT scan images. Three separate UNet models, one for liver segmentation and the others two for tumor segmentation from the segmented liver and directly from the abdominal CT scan image were used. In [24], they proposed 3D MCG-FRN + ACM is trained using the 110 cases in the LiTS dataset and evaluated on a public liver tumor dataset of the 3DIRCADb. The model in [25] realizes the segmentation framework combining deep feature with multi-scale semantic feature. In order to improve the generalization ability and training accuracy of the model, this paper proposes a combination of the traditional multi-classification cross-entropy loss function with the content loss function of generator output and the adversarial loss function of discriminator. [26] proposes a novel network named Multi-scale Attention Net (MA-Net) by introducing self-attention mechanism into our method to adaptively integrate local features with their global dependencies. [27] presents a liver tumor CT image segmentation method based on AI medical imaging-based technology. This study proposed an artificial intelligence-based *K*-means clustering (KMC) algorithm. In [28], the model comprised of three parts: synthetic image generation, localization, and segmentation, is proposed. An optimized generative adversarial network (GAN) is utilized for generation of synthetic images. The generated images are localized by using the improved localization model, in which deep features are extracted from pre-trained Resnet-50 models and fed into a YOLOv3 detector as an input. The proposed modified model localizes and classifies the minute liver tumor with 0.99 mean average precision (mAp). The third part is segmentation. In [29], U^n -Net, an n-fold network architecture, was proposed based on the traditional U-Net. In the U^n -Net model, the output features of the convolution units are taken as the skip connection. In [30], a novel framework for automated liver segmentation via a level set formulation is presented. A sparse representation of both global (region-based) and local (voxel-wise) image information is embedded in a level

set formulation to innovate a new cost function. The network in [31] is one that segments the lesions consists of a cascaded architecture, which first focuses on the region of the liver in order to segment the lesions on it. [32] proposes a method for anatomical data augmentation that is based on using slices of computed tomography (CT) examinations that are adjacent to labeled slices as another resource of labeled data for training the network. The paper in [33] proposes a method for automatic segmentation of liver lesions using deep residual networks (ResNet). [34] describes a method for automatically segmenting the liver and its lesions in CT abdomen images using cascaded fully convolutional neural networks (CFCNs) and dense 3D conditional random fields (CRFs). This paper in [35] presents a method that employs cascaded fully convolutional neural networks (CFCNs) for the segmentation of the liver and its lesions in CT and MRI abdomen images. Two FCNs are trained and cascaded for liver and lesion segmentation, where the first FCN segments the liver, and the second FCN focuses on lesions within the predicted liver regions. The paper in [36] presents a deep convolutional neural network (DCNN) method for liver lesion segmentation. The DCNN model operates in 2.5D, taking a stack of adjacent slices as input and generating a segmentation map for the center slice. The model comprises 32 layers and combines long-range concatenation connections from U-Net and short-range residual connections from ResNet. [37] presents an automatic method for liver tumor segmentation from computed tomography (CT) images using convolutional neural networks (CNNs). CNNs are employed as deep learning models with convolutional filters to learn hierarchical features from the data. The CNNs model is compared to traditional machine learning algorithms, including AdaBoost, Random Forests (RF), and support vector machine (SVM). The paper in [38] introduces a novel hybrid densely connected UNet (H-DenseUNet) for liver and tumor segmentation in medical images. This method combines a 2D DenseUNet for efficient intra-slice feature extraction and a 3D counterpart for aggregating volumetric contexts. The paper in [39] presents deep learning algorithm with graph cut refinement for automatic liver segmentation in CT scans. The method comprises two main steps: (i) simultaneous liver detection and probabilistic segmentation using a 3D convolutional neural network, and (ii) accuracy refinement of the initial segmentation. A fully automatic approach for liver tumor segmentation from CT images using a multi-channel fully convolutional network (MC-FCN). This method employs different networks for each phase of the CT images to capture distinct pathological features, and their high-layer features are fused together. This is from [40]. [41] presents a hierarchical framework based on deep fully convolutional-deconvolutional neural networks (CDNN) for liver and liver tumor segmentation. It involves multiple CDNN models trained for different segmentation tasks. [42] proposes an automatic and efficient algorithm for liver segmentation from 3D CT volumes. They use a deep image-to-image network (DI2IN) that employs a convolutional encoder-decoder architecture combined with multi-level feature concatenation and deep supervision. [43] includes several steps such as thresholding, median filtering, the application of the GCC (Greatest Connected Component) algorithm, morphological filtering, edge detection using Sobel filters, and the superimposition of contours on the original image to obtain the segmented result. The proposed method in [44] utilizes a hybrid ResUNet model, which combines the ResNet and UNet models. This approach is used for segmenting the liver and assessing the region of interest (ROI) in CT image volumes. The chapter in [45] presents a survey of medical image segmentation techniques based on the features used by each method. It categorizes these methods into three groups. [46] conducts a comparative study between 10 automatic and six interactive methods for liver segmentation from contrast-enhanced CT images. The study includes algorithms such as statistical shape models, atlas registration, level-sets, graph-cuts, and rule-based systems. The method in [47] consists of several stages: Automatically delineating skin, bones, lungs, kidneys, and spleen using thresholding, mathematical morphology, and distance maps. The study in [48] focused on patients who received a liver transplant for chronic HCV infection, had recurrent HCV infection after transplantation, and subsequently achieved a sustained virologic response (SVR) to anti-HCV therapy. An

improved three-dimensional (3D) U-Net network was proposed in [49] for liver segmentation from abdominal CT images. A semi-supervised 3D liver segmentation optimization algorithm was developed by embedding the 3D U-Net network into a generative adversarial network (GAN). Deep convolutional neural networks (DCNN) based on a feature restoration method were used to generate realistic fake images for the training process. The method employs convolutional neural networks (CNN) based on the 2D U-Net architecture and a random forest classifier. The method in [50] employs convolutional neural networks (CNN) based on the 2D U-Net architecture and a random forest classifier. The proposed algorithm in [51], named "MD-ASM," employs an active shape model (ASM) and introduces a new feature-constrained Mahalanobis distance cost function to measure the distance between the generated shape during the iterative step and the mean shape model. [52] explains a three-level AdaBoost-guided active shape model for liver segmentation in CT images. [53] employs a fully automatic 3D liver segmentation framework for challenging cases in abdominal CT images. The paper in [54] uses a cascaded classifier framework based on deep convolutional neural networks for liver tumor segmentation in CT abdominal images. [55] employs a Three-dimensional dual path multiscale convolutional neural network (TDP-CNN) for liver and liver tumor segmentation from CT images, with refinement using conditional random fields (CRF). Paper [56] uses deep learning-based methods, including convolutional neural networks and fully convolutional networks, applied to the analysis of focal liver lesions, parenchyma, and the vascular system. The presented framework in [57] aims to improve liver segmentation in portal phase abdominal CT images, especially in challenging cases with low contrast with neighboring organs, the presence of pathologies, and highly varied liver shapes between subjects. The framework includes a two-step process. Paper [58] shows a deep learning-based liver segmentation algorithm used for image fusion in CT-guided needle placement procedures. And finally paper [59] explains a hierarchical framework based on deep fully convolutional-deconvolutional neural networks (CDNN) for liver and liver tumor segmentation in contrast-enhanced abdominal CT images.

III METHODOLOGY

Data and Pre processing

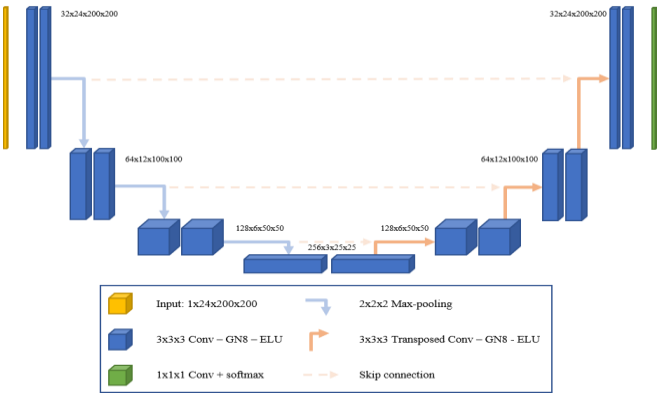
The adjustData function takes as input an image (img), a corresponding mask (mask), a boolean flag indicating whether it's a multi-class segmentation task (flag_multi_class), and the number of classes (num_class).
If it's a multi-class task, the function normalizes the image values to be in the range [0,1] and converts the mask into a one-hot encoded format. The one-hot encoding is done by creating a new mask array (new_mask) where each channel corresponds to a different class, and the pixel values are set to 1 for the class that the original pixel belonged to.
If it's not a multi-class task, the function checks if the image values are in the range [0,1] and normalizes them if needed. It then threshold the mask values to convert them into binary values (0 or 1).

The trainGenerator function is a data generator for training a neural network. It generates batches of augmented images and corresponding masks.
It uses two instances of ImageDataGenerator from Keras (image_datagen and mask_datagen) to perform data augmentation on the images and masks.
The function then uses the flow_from_directory method to generate batches of images and masks from the specified directories (train_path, image_folder, and mask_folder). It allows the function to read images and masks directly from directories and apply data augmentation on the fly.
The train_generator is created by zipping together the image and mask generators, and for each batch, the adjustData function is called to preprocess the images and masks before yielding them. The generator is designed to be used in training deep learning models for image segmentation.

```
In [6]: def adjustData(img,mask,flag_multi_class,num_class):
            if(flag_multi_class):
                img = img / 255
                mask = mask[:,:,:,0] if(len(mask.shape) == 4) else mask[:,:,0]
                new_mask = np.zeros(mask.shape + (num_class,))
                for i in range(num_class):
                    #for one pixel in the image, find the class in mask and convert it into one-hot vector
                    #index = np.where(mask == i)
                    #index_mask = (index[0],index[1],index[2],np.zeros(len(index[0]),dtype = np.int64) + i) if (len(mask.shape) == 4) els
                    #new_mask[index_mask] = 1
                    new_mask[mask == i,i] = 1
                new_mask = np.reshape(new_mask,(new_mask.shape[0],new_mask.shape[1]*new_mask.shape[2],new_mask.shape[3])) if flag_multi_c
                mask = new_mask
            elif(np.max(img) > 1):
                img = img / 255
                mask = mask /255
                mask[mask > 0.5] = 1
                mask[mask <= 0.5] = 0
            return (img,mask)
        def trainGenerator(batch_size,train_path,image_folder,
                           mask_folder,
                           aug_dict,
                           image_color_mode = "grayscale",
                           mask_color_mode = "grayscale",
                           image_save_prefix  = "image",
                           mask_save_prefix  = "mask",
                           flag_multi_class = False,
                           num_class = 2,
                           save_to_dir = None,
                           target_size = (256,256),
                           seed = 1,
                           subset = 'training'):
            image_datagen = ImageDataGenerator(**aug_dict)
            mask_datagen = ImageDataGenerator(**aug_dict)
            image_generator = image_datagen.flow_from_directory(
                train_path,
                classes = [image_folder],
                class_mode = None,
                color_mode = image_color_mode,
                target_size = target_size,
                batch_size = batch_size,
                save_to_dir = save_to_dir,
                save_prefix  = image_save_prefix,
                seed = seed,
                subset = subset)
            mask_generator = mask_datagen.flow_from_directory(
                train_path,
                classes = [mask_folder],
                class_mode = None,
                color_mode = mask_color_mode,
                target_size = target_size,
                batch_size = batch_size,
                save_to_dir = save_to_dir,
                save_prefix  = mask_save_prefix,
                seed = seed,
                subset = subset)
            train_generator = zip(image_generator, mask_generator)
            for (img,mask) in train_generator:
                img,mask = adjustData(img,mask,flag_multi_class,num_class)
                yield (img,mask)
```

Architecture

We make use of the Unet architecture in this project. UNet is a popular architecture for image segmentation tasks because it is able to achieve high accuracy while being computationally efficient.
It is a fully convolutional network, meaning that it only uses convolutional layers and does not have any fully connected layers. This makes UNet well-suited for image segmentation tasks because it can preserve spatial information throughout the network.
UNet consists of two main paths: a contracting path and an expansive path. The contracting path consists of a series of convolutional and pooling layers that reduce the spatial resolution of the input image. The expansive path consists of a series of transposed convolutional and upsampling layers that increase the spatial resolution of the feature maps.
At each level of the network, the contracting and expansive paths are connected by skip connections. These skip connections allow the network to learn precise localization of features at different scales.



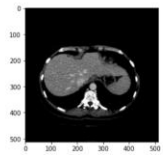
Implementation

Libraries

```
In [1]: from tensorflow.keras.preprocessing.image import ImageDataGenerator
        import numpy as np
        import tensorflow as tf
        import pandas as pd
        from tqdm import tqdm
        import os
        from cv2 import imread, createCLAHE
        import cv2
        from glob import glob
        %matplotlib inline
        import matplotlib.pyplot as plt
        from IPython.display import clear_output
        from tensorflow.keras.optimizers import Adam
        from sklearn.model_selection import train_test_split

In [2]: train_imgs_path = "C:/Users/SOUBITH/Desktop/Soubith_VIT/VIT 7th Sem/Medical Image Analysis/Jcomp/3DIRCADB/train/Images"
        train_masks_path = "C:/Users/SOUBITH/Desktop/Soubith_VIT/VIT 7th Sem/Medical Image Analysis/Jcomp/3DIRCADB/test/Masks"
        test_imgs_path = "C:/Users/SOUBITH/Desktop/Soubith_VIT/VIT 7th Sem/Medical Image Analysis/Jcomp/3DIRCADB/test/Images"
        test_masks_path = "C:/Users/SOUBITH/Desktop/Soubith_VIT/VIT 7th Sem/Medical Image Analysis/Jcomp/3DIRCADB/test/Masks"
```

Dataset

```
In [3]: img = cv2.imread("C:/Users/SOUBITH/Desktop/Soubith_VIT/VIT 7th Sem/Medical Image Analysis/Jcomp/3DIRCADB/test/Images/images/image
        plt.imshow(img)
        print(img.shape, np.amin(img), np.amax(img))

(512, 512, 3) 0 255
```



UNet model

```
In [5]: from tensorflow.keras.models import *
        from tensorflow.python.keras.utils.data_utils import Sequence
        from tensorflow.keras.layers import *
        from tensorflow.keras.optimizers import *
        from tensorflow.keras import backend as keras
        from tensorflow.keras.preprocessing.image import ImageDataGenerator
        from tensorflow.keras.callbacks import ModelCheckpoint, LearningRateScheduler


        def dice_coef(y_true, y_pred):
            y_true_f = keras.flatten(y_true)
            y_pred_f = keras.flatten(y_pred)
            intersection = keras.sum(y_true_f * y_pred_f)
            return (2. * intersection + 1) / (keras.sum(y_true_f) + keras.sum(y_pred_f) + 1)

        def dice_coef_loss(y_true, y_pred):
            return -dice_coef(y_true, y_pred)

        def unet(input_size=(256,256,1)):
            inputs = Input(input_size)

            conv1 = Conv2D(32, (3, 3), activation='relu', padding='same')(inputs)
            conv1 = Conv2D(32, (3, 3), activation='relu', padding='same')(conv1)
            pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)

            conv2 = Conv2D(64, (3, 3), activation='relu', padding='same')(pool1)
            conv2 = Conv2D(64, (3, 3), activation='relu', padding='same')(conv2)
            pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)

            conv3 = Conv2D(128, (3, 3), activation='relu', padding='same')(pool2)
            conv3 = Conv2D(128, (3, 3), activation='relu', padding='same')(conv3)
            pool3 = MaxPooling2D(pool_size=(2, 2))(conv3)

            conv4 = Conv2D(256, (3, 3), activation='relu', padding='same')(pool3)
            conv4 = Conv2D(256, (3, 3), activation='relu', padding='same')(conv4)
            pool4 = MaxPooling2D(pool_size=(2, 2))(conv4)
```

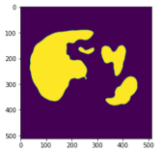
```
    pool4 = MaxPooling2D(pool_size=(2, 2))(conv4)

    conv5 = Conv2D(512, (3, 3), activation='relu', padding='same')(pool4)
    conv5 = Conv2D(512, (3, 3), activation='relu', padding='same')(conv5)

    up6 = concatenate([Conv2DTranspose(256, (2, 2), strides=(2, 2), padding='same')(conv5), conv4], axis=3)
    conv6 = Conv2D(256, (3, 3), activation='relu', padding='same')(up6)
    conv6 = Conv2D(256, (3, 3), activation='relu', padding='same')(conv6)

    up7 = concatenate([Conv2DTranspose(128, (2, 2), strides=(2, 2), padding='same')(conv6), conv3], axis=3)
    conv7 = Conv2D(128, (3, 3), activation='relu', padding='same')(up7)
    conv7 = Conv2D(128, (3, 3), activation='relu', padding='same')(conv7)

    up8 = concatenate([Conv2DTranspose(64, (2, 2), strides=(2, 2), padding='same')(conv7), conv2], axis=3)
    conv8 = Conv2D(64, (3, 3), activation='relu', padding='same')(up8)
    conv8 = Conv2D(64, (3, 3), activation='relu', padding='same')(conv8)

    up9 = concatenate([Conv2DTranspose(32, (2, 2), strides=(2, 2), padding='same')(conv8), conv1], axis=3)
    conv9 = Conv2D(32, (3, 3), activation='relu', padding='same')(up9)
    conv9 = Conv2D(32, (3, 3), activation='relu', padding='same')(conv9)

    conv10 = Conv2D(1, (1, 1), activation='sigmoid')(conv9)

    return Model(inputs=[inputs], outputs=[conv10])
```

Compiling and fitting the model

```
In [13]: model.compile(optimizer=Adam(learning_rate=1e-5), loss=dice_coef_loss,
                      metrics=[dice_coef, 'binary_accuracy'])
         # model.compile(optimizer = Adam(lr = 1e-4), loss = 'binary_crossentropy', metrics = ['accuracy'])

In [14]: from tensorflow.keras.callbacks import ModelCheckpoint, LearningRateScheduler, EarlyStopping, ReduceLROnPlateau
         weight_path="{}_weights.best.hdf5".format('cxr_reg')

         checkpoint = ModelCheckpoint(weight_path, monitor='val_loss', verbose=1,
                                      save_best_only=True, mode='min', save_weights_only = True)

         reduceLROnPlat = ReduceLROnPlateau(monitor='val_loss', factor=0.5,
                                            patience=3,
                                            verbose=1, mode='min', epsilon=0.0001, cooldown=2, min_lr=1e-6)
         early = EarlyStopping(monitor="val_loss",
                               mode="min",
                               patience=15) # probably needs to be more patient, but kaggle time is limited
         callbacks_list = [checkpoint, early]

In [15]: num_train = 1647
         num_valid = 411
         train_batchsize = 16
         valid_batchsize = 16
         Step_train = num_train // train_batchsize
         Step_valid = num_valid // valid_batchsize
         epochs = 100

In [16]: history = model.fit(
             train_generator,
             steps_per_epoch=Step_train,
             epochs=epochs,
             validation_data=valid_generator,
             validation_steps=Step_valid,
             verbose=1,
             callbacks=callbacks_list
         )
```

```
102/102 [==============================] - 128s 1s/step - loss: -0.0782 - dice_coef: 0.0782 - binary_accuracy: 0.6895 - val_l
oss: -0.0648 - val_dice_coef: 0.0650 - val_binary_accuracy: 0.7346

Epoch 00001: val_loss improved from inf to -0.06482, saving model to cxr_reg_weights.best.hdf5
Epoch 2/100
102/102 [==============================] - 94s 930ms/step - loss: -0.0895 - dice_coef: 0.0895 - binary_accuracy: 0.6615 - val
_loss: -0.0895 - val_dice_coef: 0.0894 - val_binary_accuracy: 0.3889

Epoch 00002: val_loss improved from -0.06482 to -0.08950, saving model to cxr_reg_weights.best.hdf5
Epoch 3/100
102/102 [==============================] - 94s 931ms/step - loss: -0.1297 - dice_coef: 0.1297 - binary_accuracy: 0.6797 - val
_loss: -0.1071 - val_dice_coef: 0.1075 - val_binary_accuracy: 0.8431

Epoch 00003: val_loss improved from -0.08950 to -0.10707, saving model to cxr_reg_weights.best.hdf5
Epoch 4/100
102/102 [==============================] - 95s 935ms/step - loss: -0.4917 - dice_coef: 0.4918 - binary_accuracy: 0.9545 - val
_loss: -0.6315 - val_dice_coef: 0.6320 - val_binary_accuracy: 0.9754

Epoch 00004: val_loss improved from -0.10707 to -0.63147, saving model to cxr_reg_weights.best.hdf5
Epoch 5/100
```

```
In [20]: results.shape
Out[20]: (765, 512, 512, 1)

In [21]: results[results >= 0.5] = 255
         results[results < 0.5] = 0
         results = np.asarray(results, np.uint8)

In [22]: plt.imshow(results[5])
Out[22]: <matplotlib.image.AxesImage at 0x7f0f905fea50>
```

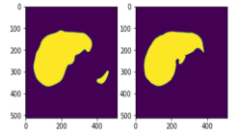


IV RESULTS

```
In [23]: img = cv2.imread("C:/Users/SOUBITH/Desktop/Soubith_VIT/VIT 7th Sem/Medical Image Analysis/Jcomp/3DIRCADB/test/Images/images/imag
         img = img/255
         image = img[:,:,0]
         image_tensor = tf.convert_to_tensor(image, dtype=tf.float32)
         image_tensor = tf.expand_dims(image_tensor, 0)
         mask = model.predict(image_tensor)
         mask[mask >= 0.5] = 255
         mask[mask < 0.5] = 0
         mask = np.asarray(mask, np.uint8)
         plt.subplot(1,2,1)
         plt.imshow(mask[0])
         plt.subplot(1,2,2)
         true_mask = cv2.imread("C:/Users/SOUBITH/Desktop/Soubith_VIT/VIT 7th Sem/Medical Image Analysis/Jcomp/3DIRCADB/test/Masks/masks/
         _true_mask = true_mask[:,:,0]
         plt.imshow(_true_mask)
```

Ảnh dự đoán và true mask:

```
Out[23]: <matplotlib.image.AxesImage at 0x7f100ec738d0>
```



Predicted and the true mask is shown.

V CONCLUSION

Liver segmentation is a crucial step in medical image analysis, playing a pivotal role in liver cancer diagnosis, treatment planning, and monitoring. UNet, a convolutional neural network (CNN) architecture, has emerged as a powerful tool for liver segmentation, demonstrating remarkable performance in various medical imaging applications.

As deep learning techniques continue to evolve, their role in liver tumor segmentation is expected to grow even more prominent. Data Augmentation; Utilizing data augmentation techniques can expand the dataset and enhance the network's generalization ability, leading to more robust segmentation performance. Transfer Learning; Leveraging transfer learning from pre-trained models can accelerate the training process and improve the network's segmentation capabilities. Multi-Task Learning; Incorporating additional tasks, such as lesion segmentation or tissue classification, can further enhance the network's overall performance.

VI REFERENCES

1. Ryu, H., Shin, S. Y., Lee, J. Y., Lee, K. M., Kang, H. J., & Yi, J. (2021). Joint segmentation and classification of hepatic lesions in ultrasound images using deep learning. *European radiology*, *31*, 8733-8742.
2. Han, X. (2017). Automatic liver lesion segmentation using a deep convolutional neural network method. *arXiv preprint arXiv:1704.07239*.
3. Christ, P. F., Elshaer, M. E. A., Ettlinger, F., Tatavarty, S., Bickel, M., Bilic, P., ... & Menze, B. H. (2016). Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random fields. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19* (pp. 415-423). Springer International Publishing.
4. Olaf Ronneberger, Philipp Fischer, and Thomas Brox, UNet: Convolutional Networks for Biomedical Image Segmentation, pp. 234–241, Springer International Publishing, Cham, 2015
5. Freiman, M., Cooper, O., Lischinski, D., Joskowicz, L.: Liver tumors segmentation from cta images using voxels classification and affinity

constraint propagation. Int J Comput Assist Radiol Surg 6(2), 247–255 (2011)
6. H¨ame, Y., Pollari, M.: Semi-automatic liver tumor segmentation with hidden markov measure field model and non-parametric distribution estimation. Med Image Anal 16(1), 140–149 (2012)
7. Kadoury, S., Vorontsov, E., Tang, A.: Metastatic liver tumour segmentation from discriminant grassmannian manifolds. Phys Med Biol 60(16), 6459 (2015)
8. Li, G., Chen, X., Shi, F., Zhu, W., Tian, J., Xiang, D.: Automatic liver segmentation based on shape constraints and deformable graph cut in ct images. IEEE Trans. Image Process. 24(12), 5315–5329 (2015)
9. Linguraru, M.G., Richbourg, W.J., Liu, J., Watt, J.M., Pamulapati, V., Wang, S., Summers, R.M.: Tumor burden analysis on computed tomography by automated liver and tumor segmentation. IEEE Trans. Med. Imag. 31(10), 1965–1976 (2012)
10. Chlebus, G., Meine, H., Moltz, J. H., & Schenk, A. (2017). Neural network-based automatic liver tumor segmentation with random forest-based candidate filtering. *arXiv preprint arXiv:1706.00842*.
11. Dawant, B. M., Li, R., Lennon, B., & Li, S. (2007). Semi-automatic segmentation of the liver and its evaluation on the MICCAI 2007 grand challenge data set. *3D Segmentation in The Clinic: A Grand Challenge*, 215-221.
12. Dou, Q., Chen, H., Jin, Y., Yu, L., Qin, J., & Heng, P. A. (2016). 3D deeply supervised network for automatic liver segmentation from CT volumes. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19* (pp. 149-157). Springer International Publishing.
13. Al-Shaikhli, S. D. S., Yang, M. Y., & Rosenhahn, B. (2015). Automatic 3D liver segmentation using sparse representation of global and local image information via level set formulation. *arXiv preprint arXiv:1508.01521*.
14. Li, X., Chen, H., Qi, X., Dou, Q., Fu, C. W., & Heng, P. A. (2018). H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes. *IEEE transactions on medical imaging*, *37*(12), 2663-2674.
15. Hu, P., Wu, F., Peng, J., Liang, P., & Kong, D. (2016). Automatic 3D liver segmentation based on deep learning and globally optimized surface evolution. *Physics in Medicine & Biology*, *61*(24), 8676.
16. Tang, Y., Tang, Y., Zhu, Y., Xiao, J., & Summers, R. M. (2020, September). E 2 Net: an edge enhanced network for accurate liver and tumor segmentation on CT scans. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 512-522). Cham: Springer International Publishing.
17. Mohamed, A. S. E. D., Salem, M. A. M., Hegazy, D., & Shedeed, H. A. (2017, September). Improved watershed algorithm for CT liver segmentation using intraclass variance minimization. In *International Conference on Information and Software Technologies* (pp. 164-176). Cham: Springer International Publishing.
18. Michael Schwier, Jan Hendrik Moltz, and Heinz-Otto Peitgen, “Object-based analysis of ct images for automatic detection and segmentation of hypodense liver lesions,” International Journal of Computer Assisted Radiology and Surgery, vol. 6, no. 6, pp. 737, 2011.
19. Schwier, M., Moltz, J. H., & Peitgen, H. O. (2011). Object-based analysis of CT images for automatic detection and segmentation of hypodense liver lesions. *International journal of computer assisted radiology and surgery*, *6*, 737-747.
20. Zhang, L., & Yu, S. C. H. (2021). Context-aware PolyUNet for liver and lesion segmentation from abdominal CT images. *arXiv preprint arXiv:2106.11330*.
21. X. Xi, L. Wang, V. S. Sheng, Z. Cui, B. Fu and F. Hu, (2020). Cascade U-ResNets for Simultaneous Liver and Lesion Segmentation. IEEE Access, 8, 68944-68952.
22. Gabriel Chartrand, Thierry Cresson et al, (2016). Liver Segmentation on CT and MR Using Laplacian Mesh Optimization. IEEE Transactions on Biomedical Engineering, 64(9), 2110-2121
23. Lakshmi Priya, B., Jayanthi, K., Pottakkat, B., & Ramkumar, G. (2020). A Modified Framework for Multislice Image Fusion for High Contrast Liver Cancer Detection. IETE Journal of Research, 66(2), 139-149.
24. Ayalew, Y. A., Fante, K. A., & Aliy, M. (2020). Deep Learning Based Liver Cancer Segmentation from Computed Tomography Images, Research Square, 2-7.
25. Z. Bai, H. Jiang, S. Li, and Y.-D. Yao, (2019).Liver tumor segmentation based on multiscale candidate generation and fractal residual network. IEEE Access, 7, 82122-82133

26. Kaijian XIA, Hongsheng YIN et al., (2017). Liver Semantic Segmentation Algorithm Based on Improved Deep Adversarial Networks in combination of Weighted Loss Function on Abdominal CT Images. IEEE Access, 7, 96349-96358.
27. Fan, T., Wang, G., Li, Y., & Wang, H. (2020). Ma-net: A multi-scale attention network for liver and tumor segmentation. *IEEE Access*, *8*, 179656-179665.
28. Liu, L., Wang, L., Xu, D., Zhang, H., Sharma, A., Tiwari, S., ... & Shah, M. A. (2021). CT image segmentation method of liver tumor based on artificial intelligence enabled medical imaging. *Mathematical Problems in Engineering*, *2021*, 1-8.
29. Amin, J., Anjum, M. A., Sharif, M., Kadry, S., Nadeem, A., & Ahmad, S. F. (2022). Liver tumor localization based on YOLOv3 and 3D-semantic segmentation using deep neural networks. *Diagnostics*, *12*(4), 823.
30. Tran, S. T., Cheng, C. H., & Liu, D. G. (2020). A multiple layer U-Net, U n-Net, for liver and liver tumor segmentation in CT. *IEEE Access*, *9*, 3752-3764.
31. Al-Shaikhli, S.D.S.; Yang, M.Y.; Rosenhahn, B. Automatic 3D liver segmentation using sparse representation of global and local image information via level set formulation.
32. Bellver, M.; Maninis, K.-K.; Pont-Tuset, J.; Nieto, X.G.; Torres, J.; Gool, L.V. Detection-Aided liver lesion segmentation using deep learning. Machine Learning 4 Health Workshop - NIPS.
33. Ben-Cohen, A.; Klang, E.; Amitai, M.M.; Goldberger, J.; Greenspan, H. Anatomical data augmentation for CNN based pixel-wise classification. In Proceedings of the IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, USA, 4–7 April 2018;
34. Bi, L.; Kim, J.; Kumar, A.; Feng, D. Automatic liver lesion detection using cascaded deep residual networks.
35. Christ, P.F.; Elshaer, M.E.A.; Ettlinger, F.; Tatavarty, S.; Bickel, M.; Bilic, P.; Rempfler, M.; Armbruster, M.; Hofmann, F.; D'Anastasi, M.; et al. Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random fields.
36. Christ, P.F.; Ettlinger, F.; Grün, F.; Elshaer, M.E.A.; Lipkova, J.; Schlecht, S.; Ahmaddy, F.; Tatavarty, S.; Bickel, M.; Bilic, P.; et al. Automatic liver and tumor segmentation of CT and MRI volumes using cascaded fully convolutional neural networks
37. Han, X. Automatic liver lesion segmentation using a deep convolutional neural network method.
38. Li, W.; Jia, F.; Hu, Q. Automatic segmentation of liver tumor in CT images with deep convolutional neural networks. J. Comput. Commun. 2015
39. Li, X.; Chen, H.; Qi, X.; Dou, Q.; Fu, C.; Heng, P. H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation From CT Volumes. IEEE Trans. Med. Imaging 2018
40. Lu, F.; Wu, F.; Hu, P.; Peng, Z.; Kong, D. Automatic 3D liver location and segmentation via convolutional neural networks and graph cut. Int. J. Comput. Assist Radiol. Surg. 2017
41. Sun, C.; Guo, S.; Zhang, H.; Li, J.; Chen, M.; Ma, S.; Jin, L.; Liu, X.; Li, X.; Qian, X. Automatic segmentation of liver tumors from multiphase contrast-enhanced CT images based on FCN. Artificial Intell. Med. 2017
42. Yuan, Y. Hierarchical convolutional-Deconvolutional neural networks for automatic liver and tumor segmentation.
43. Yang, D.; Xu, D.; Zhou, S.K.; Georgescu, B.; Chen, M.; Grbic, S.; Metaxas, D.N.; Comaniciu, D. Automatic liver segmentation using an adversarial image-to-image network.
44. Zayene, O.; Jouini, B.; Mahjoub, M.A. Automatic liver segmentation method in CT images
45. A Deep Learning Approach for Liver and Tumor Segmentation in CT Images Using ResUNet
46. Hameedur Rahman,Tanvir Fatima Naik Bukht, Azhar Imran, Junaid Tariq,Shanshan Tu, and Abdulkareeem Alzahrani5
47. Salem, M.A.-M.; Atef, A.; Salah, A.; Shams, M. Recent survey on medical image segmentation. Comput. Vis. Concepts Methodol. Tools Appl. 2018
48. Heimann, T.; van Ginneken, B.; Styner, M.A.; Arzhaeva, Y.; Aurich, V.; Bauer, C.; Beck, A.; Becker, C.; Beichel, R.; Bekes, G.; et al. Comparison and evaluation of methods for liver segmentation from CT datasets. IEEE Trans. Med. Imaging 2009
49. Soler L., Delingette H., Malandain G., Montagnat J., Ayache N., Koehl C., Dourthe O., Malassagne B., Smith M., Mutter D., et al. Fully automatic anatomical, pathological, and

functional segmentation from CT scans for hepatic surgery.
50. Whitcomb, E.; Choi, W.T.; Jerome, K.R.; Cook, L.; Landis, C.; Ahn, J.; Westerhoff, M. Biopsy Specimens From Allograft Liver Contain Histologic Features of Hepatitis C Virus Infection After Virus Eradication. Clin. Gastroenterol. Hepatol. 2017
51. Three-Dimensional Liver Image Segmentation Using Generative Adversarial Networks Based on Feature Restoration
52. Runnan He1 Shiqi Xu2 Yashu Liu2 Qince Li1,2 Yang Liu2 Na Zhao3 Yongfeng Yuan Henggui Zhang

Chlebus, G.; Meine, H.; Moltz, J.H.; Schenk, A. Neural network-Based automatic liver tumor segmentation with random forest-Based candidate filtering.
53. 3D automatic liver segmentation using feature-constrained Mahalanobis distance in CT images.

Salman Al-Shaikhli SD, Yang MY, Rosenhahn B.
54. Fast automatic 3D liver segmentation based on a three-level AdaBoost-guided active shape model

Baochun He 1, Cheng Huang 1, Gregory Sharp 2, Shoujun Zhou 1, Qingmao Hu 1, Chihua Fang 3, Yingfang Fan 3, Fucang Jia 1
55. Shape-intensity prior level set combining probabilistic atlas and probability map constrains for automatic liver segmentation from abdominal CT images
56. Jinke Wang 1 2, Yuanzhi Cheng 3, Changyong Guo 1, Yadong Wang 1, Shinichi Tamura 4

Cascaded deep convolutional encoder-decoder neural networks for efficient liver tumor segmentation
57. Ümit Budak 1, Yanhui Guo 2, Erkan Tanyildizi 3, Abdulkadir Şengür 4

Liver tumor segmentation based on 3D convolutional neural network with dual scale
58. Lu Meng 1, Yaoyu Tian 1, Sihang Bu 1

Deep learning for image-based liver analysis - A comprehensive review focusing on malignant lesions
59. Shanmugapriya Survarachakan 1, Pravda Jith Ray Prasad 2, Rabia Naseem 3, Javier Pérez de Frutos 4, Rahul Prasanna Kumar 5, Thomas Langø 4, Faouzi Alaya Cheikh 3, Ole Jakob Elle 2, Frank Lindseth 6

Shi, C.; Cheng, Y.; Liu, F.; Wang, Y.; Bai, J.; Tamura, S. A hierar
local region-Based sparse shape composition for liver segmen
in CT scans. Pattern Recognit. 2015.
60. Deep learning-based liver segmentation for fusion-guided intervention
61. Xi Fang 1, Sheng Xu 2, Bradford J Wood 2, Pingkun Yan 3

Yuan, Y. Hierarchical convolutional-Deconvolutional neural net
for automatic liver and tumor segmentation.