

Tensor Flow Model in Medical Image Analysis - Review

Raswitha Bandi^a, D. Dakshayani Himabindu^b, Sahiti Cheguru^c, S. Ajay Kumar^d

^aAssistant Professor, Department of Information Technology, VNR Vignan Jyothi Institute of Engineering and Technology, Bachupally, Hyderabad, India.

^bAssistant Professor, Department of Information Technology, VNR Vignan Jyothi Institute of Engineering and Technology, Bachupally, Hyderabad, India.

^cUG Scholar, Department of CSE, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India.

^dAssociate Professor, Department of Computer Science and Engineering, Malla Reddy Engineering College (Autonomous), Hyderabad, India.

Article History: Received: 11 January 2021; Accepted: 27 February 2021; Published online: 5 April 2021

Abstract: Support Vector Machines, Reinforcement algorithms, artificial neural networks are some of the Machine Learning Algorithms available in Medical Analysis. By using these algorithms, much of the research has been done in analysis of liver cancer for genome classification and identification of lesions. At present, Deep learning algorithms have quickly turned into a strategy for examine CT images. This article presents one of the major deep learning techniques named tensor flow technique to investigate images in scan for the task of visualization of abnormal condition of liver tumor in the context of shape and color towards disease diagnosis. We surveyed the utilization of tensor flow for classifying images, detection of objects, and detection of lesions. In this paper, we mainly concentrated on the study and working of tensor flow in image classification. Also, a summary of the present and future scope in this area has been presented in detail.

Keywords: Artificial Neural Network, Deep Learning, Medical Image Analysis, Tensor Flow, Liver, Lesion.

1. Introduction

Deep learning which is also called as Deep Structured Learning or Hierarchical learning basically is a part of Machine Learning methods. It is in view of learning information portrayals instead of undertaking particular calculations [1]. In this Practice, the learning can be regulated, incompletely managed, and unsupervised. Deep Structured learning can be characterized as another region of machine picking up attempting to enhance territories like voice inquiry, picture and dialect preparation or understand unstructured information challenges. Deep learning can empower us to build our computerized promoting exercises by changing the substance of our sites and versatile applications into tailor made offers [2]. The fame of Deep learning started in March 2016, when Google's Deep Mind AI program called Alpha Go bested Lee Sedol, the commended player of the table game "Go", by winning four out of five amusements [1,7,8]. After the match it was uncovered that a generally new AI strategy called "deep learning" was in charge of the triumph. As indicated by researchers, profound learning innovation can possibly change the whole AI range [1] [8]. Deep structured learning is a part of AI that includes [1] [8]:

- 1) Utilize the course of various network layers of unordered preparing segments for characteristic extrication and change. It retrieves the output from the past layer as information.
- 2) Acquire Knowledge of directed (Ex: grouping) and additionally unsupervised (Ex: design examination) behaviors.
- 3) Acquire Knowledge on different portrayals that relate to various sizes of reflection; the levels frame a chain of command of ideas.
- 4) Use some kind of slope drop for preparing by means of back propagation.

Deep Learning can utilize the layers and it includes shrouded layers of dummy neural system and contains propositional formulae. They may incorporate dormant factors sorted out the layers in significant generative models such as the nodes in Deep Belief networks and Deep Boltzmann Machines [1, 8].

1.1. Working Procedure of Deep Learning

Deep Learning includes encouraging a computer system to give a considerable measure of information this will make good use of choices about other information. The information is nourished using systems, similar to the machine learning use case. The networks are logical structures and they are the sequence of actual/fake

questions, and the outcome of these questions is a numerical value, of each line of information which bypasses thru these networks, and after that receives the answers according to classification.

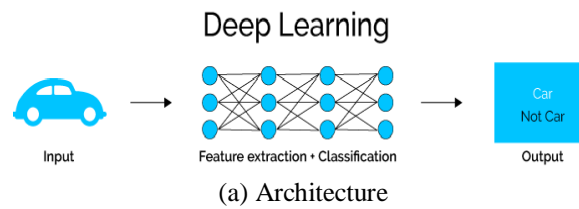


Figure 1. Structure of Deep Learning

1.2. Applications of Deep Learning

Some of the applications of deep learning include [8]:

- 1) Black and white images colorization.
- 2) Sound effects in silent movies.
- 3) Programmed Machine Translation.
- 4) Classification of objects in Photographs.
- 5) Generation of automatic handwriting.
- 6) Generation of characters in text.
- 7) Generation of captions for image.
- 8) Automatic game playing.

2. Deep Learning Frameworks

Several deep learning frameworks are available and the top most deep learning frameworks are: tensor flow, Theano, pytorch, deeplearning4j, and caffe. In this paper we are mainly concentrating on one of the top most deep learning frameworks called tensor flow.

2.1. Introduction to Tensor Flow

Tensor Flow is an open-source programming library for dataflow programming over a scope of undertakings. It is an emblematic math library, and is utilized for machine learning applications such as neural systems. Releasing the Tensor Flow AI engine publicly, with regards to AI, the genuine esteem lies less in the product or the calculations as the information expected to make it is smarter. With deep learning, we can instruct frameworks to perform assignments such as perceiving pictures, distinguishing talked words, and notwithstanding understanding characteristic dialect by nourishing information into huge neural systems associated machines that surmised the web of neurons inside the human mind.

2.2. Applications of Tensor Flow

The essential Primary tool of Deep learning is Tensor Flow [3][5]. It is an open source computerized reasoning library, utilizing information stream charts to construct models. It enables designers to make expansive systems with so many network layers. Tensor Flow is utilized for Classification of images, image Perception, understanding of data, Discovering the images, Prediction and Creation of data [3][5].

2.2.1. Principle Use Cases of Tensor Flow

2.2.1.1. Voice/Sound Recognition

A standout amongst the most surely understood employments of Google Tensor Flow are voice or Sound based applications [3]. The best possible information encourage; neural systems are equipped for understanding sound signs. These can be:

- Voice acknowledgment – for the most part utilized as a part of IoT, Automotive, Security and UX/UI[43][44].
- Voice seeks – for the most part utilized as a part of Telecoms and Handset Manufacturers.
- Sentiment Analysis – for the most part utilized as a part of CRM.
- Flaw Detection (engine noise) – for the most part utilized as a part of Automotive and Aviation.

In order to utilize cases [2], we are generally acquainted with sound-inquiry and sound-initiated collaborators are with advanced mobile phones. For example, Apple's Siri, Google Now for Android and Microsoft Cortana for Windows Phone [3][5].

Regional Language interpretation is another regular use case for sound Recognition. Discourse to content applications can be utilized to decide the bits of sound in more prominent sound records and decipher the talked word as content[3][5]. Sound based applications can be utilized as a part of CRM.

2.2.1.2. Text based Applications

The class of text based applications are content based applications, for example, investigation of social media (CRM, Social Media), Threat Detection (Social Media, Government) and Fraud Detection (Insurance, Finance) [3][5]. Regional language interpretation is a standout amongst other content based applications.

Google Translate, which bolsters more than 100 dialects making an interpretation starting with one and then the next [3][15].

- Text Summarization
- Smart Reply

2.2.1.3. Image Recognition

This is widely utilized by Social networks, telecommunications industry and handheld device makers [2][3][16]. Facial image recognition, searching of images, Motion picture detection can also be used in Automotive, Aviation and Health care industry. The main aim of image recognition is recognizing & classifying objects and people in CT images and analyzes the content and context. In the context of medical image analysis, tensor flow object recognition algorithms [2][7][8] classify and indicate subjective objects within the larger images. These tensor flow algorithms can process vast information and identify more patterns. After that systems are able to review CT-scans and identify illness more effectively compared to human beings.

2.2.1.4. Time Series

Tensor Flow Time Series calculations are utilized for dissecting time arrangement information so as to separate significant insights. Generally time series calculations are used to estimate the non-regular times and to produce other time series version [3] [5][17]. Widely recognized use case for Time Series is exhortation. We know about this utilization from Amazon, Google, Face book and Netflix where they break down client movement and contrast it with the a huge number of different clients to figure out what the client may get a kick out of the chance to buy or watch. Alternate uses of Tensor Flow Time Series calculations are in the field of Finance, Accounting, Government, Security and IoT with Risk Detections, Predictive Analysis and Enterprise/Resource Planning [2][3][18].

2.2.1.5. Video Detection

Tensor Flow neural systems can use video data for detection of motion of pictures, real time drift detection in gaming, and to provide security to the data [3][5]. For example, YouTube-8M is proposing to animate a thesis on sweeping large scale video data streaming, representation learning, noisy data illustrating, trade learning, and

space selection[19].In this paper we are mainly concentrating on one of the important use cases of Tensor flow called object recognition that is used to detect, classify and recognize lesions in the liver. Here, in this Paper I introduced one of the use cases of medical image analysis i.e. Liver Cancer.

3. Liver Cancer Introduction

[4]Liver cancer is one of the most regular sorts of destructive infections, demonstrating in charge of the passing's of 600,000 patients worldwide in 2001 alone [2]. The frequency of liver lesions is considerably larger than the other classification of cancers, like colorectal, lung and breast cancer, has a tendency to metastasize into the liver. Processed tomography (CT) pictures, procured after intravenous infusion of a difference specialist, are generally utilized by clinicians for the analysis, treatment and observing the liver tumors. These strategies, notwithstanding, require data about size, shape and exact area of the tumors, for which their division is essential [2] [4][20]. As the liver can extend crosswise over 150 cuts in a CT picture and contain up to many injuries, manual division is repetitive and restrictively tedious for a clinical setting. Automatic detection is also an exceptionally difficult task.

The disadvantage of manual detection and division of lesions in liver [2] [6] is that a human eye cannot predict exact tumor in liver by seeing the images. So automated tools are used to find out the lesions in liver using convolution neural networks by observing images from the CT scan. A pathologist's report subsequent to surveying a patient's natural tissue tests regularly results in effective conclusions of numerous sicknesses. A pathologist's conclusion profoundly affects a patient's treatment. Exploring of pathology slides is an extremely complex assignment, requiring a long time to pick the skill up and mastering it [10] [14][21]. Indeed, after investigation there might be mild fluctuations in the findings explored by various genetics after the investigation done by the doctors and they cannot identify the something went wrong in the findings which can lead to misdiagnoses. For instance, a few types of cancers are relatively low and their growth rate is less than forty eight percent, and this growth rate is even low for prostate cancer. The detailed diagnosis has been done by investigating the amount of information is collected from the reports. Doctors are in charge of surveying all the organic tissues on a slide [10][14]. Nonetheless, there can be many slides per tolerant, each of which is 10+ giga pixels when digitized at 40X amplification. Envision going through a thousand 10 megapixel (MP) photographs, and being in charge of each pixel [10]. Obviously, this is a considerable measure of information to cover, and frequently time is constrained. To find out Lesions in liver cancer first we have to classify the images and recognize the objects that are collected from the CT scan and then apply distributed tensor flow[5][10].

3.1. Applying Distributed Tensor Flow to Medical Data Analysis

Tensor Flow is an open-source library for a machine and a profound discovery that was initially created by Google. It contains dataflow graphs, with nodes in the diagrams speaking to mathematical operations, and the edges of the diagrams speaking to multi-dimensional information clusters (i.e., tensors) [22]. Tensor Flow has a low-level API written in Python and C++. It gives us "the capacity to modify neural system layers which in turn gives us incredible adaptability". Tensor flow utilizes GPUs, which support this as an approach to meet the overwhelming handling requests of high-stack projects. In the case of liver cancer lesion detection tensor flow uses fully convolution neural networks and foe object detection in CT-images.

Convolution Neural Networks (CNNs)

From the decades deep neural networks faced many problems to find patterns in the images, such as computer vision and voice recognition. To meet these essential components special kind of neural networks called CNNs are used [2][3][9][10].CNNs are the best display design technique for image classification [9][10][23].Convolution neural networks apply a progression of channels to the image information of a picture to separate and extract similarities in images and then they are used for characterization. Convolution neural networks contain three segments:

Convolution layers (first layer) [5][6], prosecute a predetermined range of convolution channels to the picture. For every sub region, the layer plays out an arrangement of scientific operations to create a solitary incentive in the output. Convolution layers at that point commonly apply a ReLU [9] actuation function to the output to bring non linearity 'sin the convolution layer.

Pooling layers (second layer) [5][6], this down samples the picture information separated by the convolution layers to lessen the dimensions and this component delineate request and diminishing preparation time. A

regularly utilized pooling calculation is max pooling. It extricates sub-regions of the component delineate (2x2-pixel tiles), keeps their greatest esteem, and disposes every single other esteem.

Dense (Fully connected) layers (third layer) [5][6][9], which classifies the features isolated by the first layer and down sampled by the second layers. In this layer, there is a correlation between nodes with the previous layer.

CNN is made out from heap of convolution modules that carry out extrication [5] [6][24]. Every part comprises of first layer looked after by a second layer. The last convolution module is trailed by at least one third layer that performs grouping. The last or third layer in a Convolution neural networks contains a solitary node for each objective category in the design (overall conceivable categories the design may anticipate), with a soft max or normalized exponential function enactment capacity to create an incentive between 0 and 1 for every node (the total of all these soft max or normalized exponential function esteems is equivalent to 1). By using soft max or normalized exponential function we can make relative estimations and find the objective class of each picture.

3.1.1. Building a Convolution Neural Network

The Tensor Flow layers module gives an abnormal state Application program interface makes it simple to develop a system. It imparts procedures to construct fully connected i.e., third layer and first layer after which activation functions are added and regularization functions are applied [5]. Now we will learn how layers are used to build a convolution neural network in steps.

Step1: Develop a procedure to categorize the objects in the training data by using the below Convolution neural networks structure [5] [6][25]:

1. Convolution layer#.
2. Pooling layer#1.
3. Convolution layer#2.
4. Pooling layer#2.
5. Dense Layer#1.
6. Dense layer#2.

The Tensor Flow layer Procedure holds techniques to make layers specified in step1 [5]:

convolution2d () function: Builds up a planar first layer. The input parameters for this function are no. of metastasize, metastasize kernel size, and enactment function [26].

maximum_pooling2d (): Develops a planar second layer utilizing the maximum pooling algorithm. The input parameters for this function are pooling metastasize size and path [5][6].

Dense func (): Builds a third layer. It accepts no. of neurons and actuation work as arguments [27].

The above strategy acknowledges a tensor as input and returns a changed tensor as yield. This makes it less demanding to associate one layer to another and take the yield from one layer-creation strategy and deliver it as contribution to another [5].

The following section plunge further into tf.layers used to create each layer

Input Layer

The strategies in the layers module for making first and second layers for planar picture data envision that data tensors will have a condition of [bch_size,img_width,img_height, carriers], characterized as [5][6][9][10][28]:

bch_size: Size of the batches that are used during training [5] [6].

Img_width: scope of the illustration pictures [5] [6].

Img_height: depth of the illustration pictures [5] [6].

Carriers: No. of colorific carriers in the illustration pictures. For colorific pictures, the quantity of carriers is 3 (R, G, and B). For monolithic pictures, only 1 carrier (BL-Black) [5][6][29].

Convolutional Layer#1

In this layer, we need to apply carries to Input Layer with ReLU enactment Function and we can utilize convolution2d () technique in the layers module to make this layer as [5][6][9]:

Convolution1=tf.layers.convolution2d (inputs, metastasize, size of kernel, padd, enactment=tf.nn.relu)

The sources of inputs contention determine our information

- Tensor and it must have bch_size, img_width, img_height and carriers.
- The metastasize contention determines the quantity of metastasizes to apply [5][6][30].
- The size of kernel estimate the measurements of the metastasize as scope and depth [5][6].

The padding contention indicates the yield tensor ought to have a similar scope and depth esteems as the information tensor [5][6][10][31].

The enactment contention indicates the enactment capacity to apply to the yield of the convolution [5][6][11][32].

Pooling Layer#1

After creating first convolution layer, we interface our first Pooling Layer to the Convolution layer#1 [5][6][33].

For this utilize the method maximum_pooling2d () in layers to develop a layer that performs max pooling [5][6]. i.e. the arguments for the pooling layer are pooling1=tf.layers.maximum_pooling2d (inputs, pooling_size, tread) [5][6]. The Inputs specify the input sensor which is the output from the first convolutional layer.

The pooling size contention indicates the measure of the maximum pooling channel as scope and depth. The tread contention indicates the extent of the tread i.e. it demonstrates the sub regions separated by the channel [5][6][34].

Convolution Layer#2 and Pooling Layer#2

We can associate a convolution layer#2 and pooling layer#2 to our Convolution neural network utilizing convolution2d() and maximum_pooling2d() methods as discussed above [5][35].

Second convolution layer which takes input tensor i.e. produced by the pooling Layer#1 (pool1). Second pooling layer takes convolutional layer2 output as information and delivering second pool as yield.

Dense Layer

Next Dense Layer is added to Convolution Neural Networks to perform Classification on the highlights removed by the convolution/pooling layers [5][6][36].

We utilize Dense () Method in layers to interface our fully connected layer as [5][6]:

Dense1=tf.layers.dense1 (inputs, units, Enactment)

The Inputs contention determines the info tensor i.e. pool2_flat [5][6].

The unit's contention determines the quantity of Neurons in the Dense layer [5][6].

The enactment contention determines the Enactment Function.

To enhance the consequences of our model

We can likewise apply dropout technique in Layers [19][37].

Logits Layer

The last layer in neural system is the Logits Layer, which restores the raw esteems for the Expectations.

Logits=tf.layers.dense1 (inputs, units)

This layer converts these raw values into two

Different formats [5] [6][38]:

The **Predicated class** for each sample, i.e.,

0 to 9

The probabilities for every conceivable

Target class.

Step2: Training and evaluating the CNN

Classifier

In the First step we coded CNN model function and the next is to Prepare and assess it [5][6][39].

- 1) Load preparing and test information: stack the test information or sample data.
- 2) Create the estimator: A tensor flow class for performing abnormal state display preparing, assessment and forecast.
- 3) Set up a logging hook: CNNs will take some amount of time to train the data so we can track progress during training by setting up logging hook.
- 4) Train the model
- 5) Evaluate the model to determine accuracy on the test data.
- 6) Run the model using Run command.

4. Conclusion

There are so many Machine Learning algorithms prepared on computed tomography volumes for classification & segmentation of liver cancer and to find lesions in liver. Our proposed overview contends with best in model. In this paper we explained how the sample data is submitted to deep learning frame work Tensor Flow for medical image analysis. We explained how distributed tensor flow can be used to apply medical image analysis in steps. As future work, we train and evaluate the test data and run the model using Tensor flow. We conclude that Tensor flow deep learning frameworks are promising devices for programmed examination of liver and its injuries in clinical schedule.

References

1. Goularas, D., & Kamis, S. (2019). Evaluation of deep learning techniques in sentiment analysis from twitter data. *In International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*, 12-17. <https://doi.org/10.1109/Deep-ML.2019.00011>
2. Wilamowski, B.M., Wu, B., & Korniak, J. (2016). Big data and deep learning. *In IEEE 20th Jubilee International Conference on Intelligent Engineering Systems (INES)*, 11-16.
3. <https://doi.org/10.1109/INES.2016.7555103>.
4. Militzer, A., Hager, T., Jäger, F., Tietjen, C., & Hornegger, J. (2010). Automatic detection and segmentation of focal liver lesions in contrast enhanced CT images. *In 20th International Conference on Pattern Recognition*, 2524-2527.
5. Ketkar, Nikhil. (2017). Introduction to Tensorflow. *Deep Learning with Python*, 159-194.
6. Abadi, M., Isard, M., & Murray, D.G. (2017). A computational model for TensorFlow: an introduction. *In Proceedings of the 1st ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, 1-7.
7. Sai, D.K., & Sahithi, K. (2020). D etection of abnormality in CCTV Footage: Computer Vision. *European Journal of Molecular & Clinical Medicine*, 7(4), 1148-1154.
8. Divanu Sameera, Vedavathi K, Durga Prasad Kavadi. (2020). *A Novel Approach for Text Classification using Recurrent Neural Networks*” at international journal of advanced science and technology (IJAST), 29(5).
9. Goularas, D., & Kamis, S. (2019). Evaluation of deep learning techniques in sentiment analysis from twitter data. *In International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)* 12-17.
10. Donghwoon Kwon, Hyunjoo Kim, Jinoh Kim, Sang C. Suh, Ikkyun Kim, Kuinam, Kim. (2019). A survey of deep learning-based network anomaly detection. *Cluster Computing*, 1-13.
11. Bandi, R., & Amudhavel, J. (2018). Object recognition using Keras with backend tensor flow. *International Journal of Engineering and Technology (UAE)*, 7(3.6), 229-233.

12. Bandi, R., & Anitha, G. (2018). Machine learning based Oozie Workflow for Hive Query Schedule mechanism. In *International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 513-517. <https://doi.org/10.1109/ICSSIT.2018.8748711>
13. Sahithi, K., Krishna Sai, D., & Sameera, D. (Ph.D.). (2020). Detection of Brain tumor using neural networks” at *European journal of molecular & clinical medicine (EJMCM)*,7(4), 1155-1161.
14. Charbonnier, J.P., Van Rikxoort, E.M., Setio, A.A., Schaefer-Prokop, C.M., van Ginneken, B., & Ciompi, F. (2017). Improving airway segmentation in computed tomography using leak detection with convolutional networks. *Medical image analysis*, 36, 52-60.
15. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.
16. Ben-Cohen, A., Klang, E., Amitai, M., & Greenspan, H. (2016, April). Sparsity-based liver metastases detection using learned dictionaries. In *IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, 1195-1198
17. SahithiKolla. Meenakshi Kolishetty. Sameera Divanu (Ph.D.). (2019). *Stability of Hopfield in Higher Dimension at IARJSET*, 6(7).
18. Brosch, T., Yoo, Y., Tang, L.Y.W., Li, D.K.B., Traboulsee, A., Tam, R. (2015). Deep convolutional encoder networks for multiple sclerosis lesion segmentation. *MICCAI 2015*. 9351, 3–11.
19. Li, W. (2015). Automatic segmentation of liver tumor in CT images with deep convolutional neural networks. *Journal of Computer and Communications*, 3(11), 146.
20. Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431-3440.
21. Roth, H.R., Lu, L., Liu, J., Yao, J., Seff, A., Cherry, K., & Summers, R.M. (2015). Improving computer-aided detection using convolutional neural networks and random view aggregation. *IEEE transactions on medical imaging*, 35(5), 1170-1181.
22. Ruskó, L., & Perényi, Á. (2014). Automated liver lesion detection in CT images based on multi-level geometric features. *International journal of computer assisted radiology and surgery*, 9(4), 577-593.
23. Setio, A.A.A., Ciompi, F., Litjens, G., Gerke, P., Jacobs, C., Van Riel, S.J., & Van Ginneken, B. (2016). Pulmonary nodule detection in CT images: false positive reduction using multi-view convolutional networks. *IEEE transactions on medical imaging*, 35(5), 1160-1169.
24. Vedaldi, A., & Lenc, K. (2015). Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, 689-692.
25. Vivanti, R., Ephrat, A., Joskowicz, L., Lev-Cohain, N., Karaaslan, O.A., & Sosna, J. (2015). Automatic liver tumor segmentation in follow-up CT scans: preliminary method and results. In *International workshop on patch-based techniques in medical imaging*, 54-61.
26. D.sameera, Gayathri V.C. (2018). *Finding an Accuracy on Legitimacy & Reputation value Based Malicious Node Detection & Removal Scheme on Cluster in MANETs”* at BVRIT.
27. Li, G., Chen, X., Shi, F., Zhu, W., Tian, J., & Xiang, D. (2015). Automatic liver segmentation based on shape constraints and deformable graph cut in CT images. *IEEE Transactions on Image Processing*, 24(12), 5315-5329.
28. Ben-Cohen, A., Klang, E., Diamant, I., Rozendorn, N., Amitai, M. M., & Greenspan, H. (2015). Automated method for detection and segmentation of liver metastatic lesions in follow-up CT examinations. *Journal of Medical Imaging*, 2(3), 034502.
29. 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 *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 415-423.
30. Milletari, F., Navab, N., & Ahmadi, S.A. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *fourth international conference on 3D vision (3DV)*, 565-571.
31. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., & Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, 675-678.
32. Chartrand, G., Cresson, T., Chav, R., Gotra, A., Tang, A., & DeGuisse, J. (2014). Semi-automated liver CT segmentation using Laplacian meshes. In *IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, 641-644.
33. Ben-Cohen, A., Diamant, I., Klang, E., Amitai, M., & Greenspan, H. (2016). Fully convolutional network for liver segmentation and lesions detection. In *Deep learning and data labeling for medical applications 77-85*.

34. López-Mir, F., González, P., Naranjo, V., Pareja, E., Morales, S., & Solaz-Minguez, J. (2015). A method for liver segmentation on computed tomography images in venous phase suitable for real environments. *Journal of Medical Imaging and Health Informatics*, 5(6), 1208-1216.
35. Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431-3440.
36. 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), 8676-8698.
37. Li, W. (2015). Automatic segmentation of liver tumor in CT images with deep convolutional neural networks. *Journal of Computer and Communications*, 3(11), 146-151.
38. Lu, F., Wu, F., Hu, P., Peng, Z., & Kong, D. (2017). Automatic 3D liver location and segmentation via convolutional neural network and graph cut. *International journal of computer assisted radiology and surgery*, 12(2), 171-182.
39. Vivanti, R., Ephrat, A., Joskowicz, L., Karaaslan, O., Lev-Cohain, N., & Sosna, J., Automatic liver tumor segmentation in follow up ct studies using convolution neural networks. In *Proceeding Patch-Based Methods in Medical Image Processing Workshop, MICCAI*, 2, 54–61.
40. Dou, Q., Chen, H., Jin, Y., Yu, L., Qin, J., & Heng, P. A. (2016, October). 3D deeply supervised network for automatic liver segmentation from CT volumes. In *International conference on medical image computing and computer-assisted intervention*, 149-157.
41. Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. *arXiv preprint arXiv:1409.1556*.
42. Heimann, T., et al. (2009). *Comparison and evaluation of methods for liver segmentation from ct datasets*, *IEEE Transactions on Medical Imaging*, 28(5). 1251-1265.
43. Deng, X., & Du, G. (2008, September). 3D segmentation in the clinic: a grand challenge II-liver tumor segmentation. In *MICCAI workshop*.
44. Madhuravani, B., & Dr. Murthy, D.S.R. (2017). A hybrid parallel hash model based on multi-chaotic maps for mobile data security. *Journal of Theoretical and Applied Information Technology*, 95(3), 661–669.