Image Recognition in Medicine

Chih-Hsien Lin, Yu-Hsuan Tsao† and Bing-Rong Wu†

†These authors contributed equally to this work.

Abstract

The paper makes introductions of U-Net and V-Net, and details on their architecture and features. Moreover, we implement the process of training a model of the prediction of whether a cell is benign or malignant. Last, we also study on the problems and prospect of image recognition.

Keywords: U-Net, V-Net, cancer prediction, image segmentation, image recognition

1 Introduction

Image recognition is a rapidly evolving field with numerous applications in the medical field. With the increasing availability of high-quality medical images, such as X-rays, CT scans, and MRIs, there is a growing demand for automated methods to analyze and interpret these images. Image recognition algorithms can identify specific features or abnormalities within an image, which can aid in the diagnosis and treatment of various medical conditions. This technology has the potential to revolutionize the way healthcare is practiced, by reducing the need for manual interpretation of medical images, and enabling faster and more accurate diagnoses. In this paper, we will explore the various applications of image recognition in medicine, including its use in diagnosis, prognosis, and treatment planning. We will also discuss the challenges and limitations of this technology, and its potential future developments.

2 Related Work

2.1 U-Net[1]

U-Net is a convolutional neural network for biomedical image segmentation. Generally speaking, it is well recognized that thousands of labeled

training samples are required for successful training of deep neural networks.[2] U-Net is to use the existing labeled samples more efficiently by performing image enhancement (i.e. increasing the number of images by rotating, randomly cropping, scaling, etc.) on the existing labeled images. And many networks are based on it, such as V-Net, 3d-UNet, , H-denseunet, and so on.[3] Also, there many research on it[4]. The architecture consists of a contracting path that captures context information and a symmetric expanding path that achieves precise localization. Due to its symmetrical U-shaped structure, this network is named U-Net. This network can be trained end-to-end for a relatively small number of image sets.

2.1.1 Architecture[5]

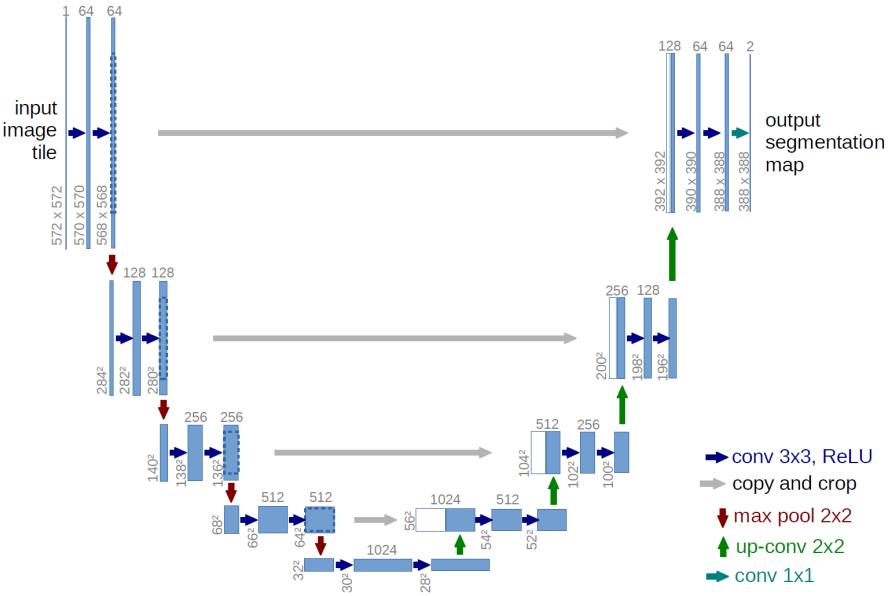

Fig. 1 This is an U-Net architecture. U-Net has 3 parts: the contracting/downsampling path, bottleneck, and the expanding/upsampling path.

2.1.2 Contracting Path

In Figure 1, the encoder part called contracting path, which is also called downsampling path. The input image tile is $572 \times 572 \times 1$ and the blue arrow is a 3×3 convolution layer. Take the top left corner, first stage, as an example, after convolution, the image tile changes from $572 \times 572 \times 1$ to $570 \times 570 \times 64$, the size shrinks by 2, and then convolution becomes $568 \times 568 \times 64$, because the network does not have any fully connected layers and only uses the valid part of each convolution, since there are actually three types of convolution operations, closely related to padding or not, and how much padding there is.

Each time the image size is doubled, the corresponding number of feature maps is also doubled. The same process is repeated four times, and by the fifth stage, at the bottom of the U-Net structure, the contracting path is completed, and the feature maps are now 28 × 28 × 1024 in size.

2.1.3 Expanding Path

After a green arrow, 2 × 2 up-sampling convolution, which is also known as inverse convolution or transpose convolution. The image size can be doubled, and correspondingly, the number of feature maps needs to be doubled to get the result in 56 × 56 × 512 data, the blue squares in the second to last layer on the right side of the diagram.

Moreover, the last two layer on the right, the forth stage of the expanding path, is a 1024-channel input of white squares and blue squares superimposed on each other. Those white squares are copied from the feature maps of the forth stage of the contracting path before they are pooled. This is the process described above, where the feature maps of the contracting path are overlaid with the feature maps of the corresponding expanded path in order to compensate for the loss of information in image size compression. Then, the 56 × 56 × 1024 input is validated twice and then follow the steps up-sampled, superimposed, and convolved. After four iterations, the output is 388 × 388 × 64.

Finally, a 1 × 1 convolution is used to compress the data into 388 × 388 × 2 because the author only needs to divide the dataset into two categories: foreground and background. A single channel can be used for dichotomous tasks, and for multiple categories, the number of corresponding categories can be modified. In summary, it can be seen that the structure of the U-net is indeed a very symmetrical U-shaped network, just like its name.

2.1.4 Overlap-tile

The input image is clearly 572 × 572, but the predicted image is 388 × 388 in Figure 1. The reason is that the author use a tip named "Overlap-tile".

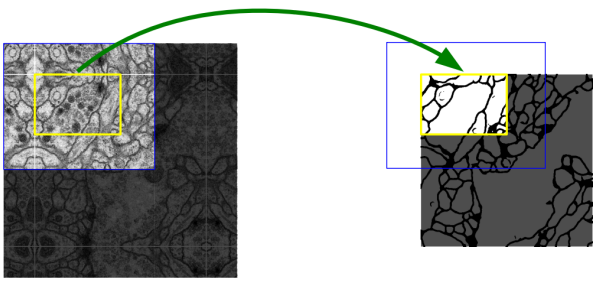

Fig. 2 Picture of Overlap-tile strategy

First of all, the actual area to be segmented is the yellow area in the figure. However, if only the yellow area is used for segmentation, it is clear that segmentation at the edges of the yellow area may be poor, as the feature information available around them is much less than that available in the middle image area. Therefore, in order to make sure of the segmentation of the edges, the actual image size of the input network should be slightly larger than the target area, which is the blue area in the figure. If the blue region is beyond the boundary of the original image, then it should not be padded with ZERO-padding, but rather with mirror image padding, which will at least ensure the integrity of the cell contours. Thus, the blue area is the network input size of 572×572, and the yellow target area is 388×388, so that when the next 388×388 prediction is completed, the blue areas of the two predictions overlap, hence the name "overlap-tile strategy".

2.1.5 Features

First, The U-Net combines the location information from the contracting path to get a general information that combines location and context, which is necessary to predict a good segmentation map.

Besides, no dense layers are used, so image size can be used.

Moreover, for irregular images, it can identify the edges and can be used for gastric ulcers, tumour locations and irregular lesion locations.

However, it is not possible to output the number of images. For instance, it is not possible to tell how many tumours are in the image, only to determine their location.

2.2 V-Net[6][7][8]

Convolutional neural networks (CNNs) have been widely used in computer vision and medical image analysis in recent years. Most methods can only deal with 2D images, while most medical data for clinical applications consist of 3D volumes.V-Net provides a three-dimensional image segmentation method, which adopts an end-to-end training method, and uses a new objective function based on Dice coefficient to optimize training during training. It works well for cases where there is a severe imbalance between the number of foreground and background voxels. To handle situations where limited data is available for training, it uses random non-linear transformations and histogram matching to augment the data.Now many medical image segmentation networks are improved based on the U-Net network, so many networks of this type have been derived, and they are all similar.

2.3 Convolutional Layer

The convolution layer is responsible for extracting local features in the image. The principle is to slide and extract features on the image through many convolution kernels (filters, kernels).The figure below shows the process of

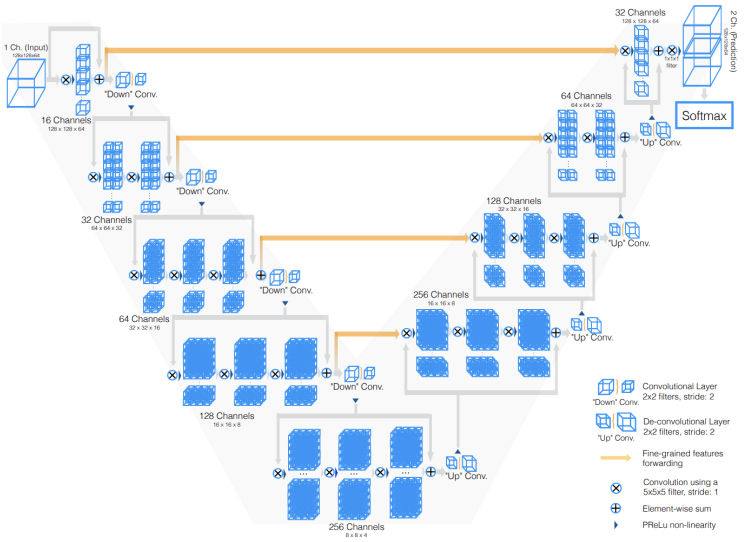

Fig. 3 The left side of the V-Net network is a contracting path; the right side is a decontracting path, in order to restore the image to its original size.

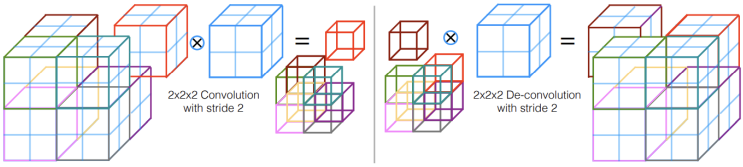

Fig. 4 This shows thee details of how the convolution operate.

convolution sliding. The left, middle and right are the input layer, convolution kernel, and output layer respectively.

2.3.1 Convolution Operation

The method of convolution operation is to multiply the sliding window and the convolution kernel point-to-point (elementwise), and then add the multiplied values.

2.3.2 Convolution

For convolution operation, the convolution kernel size we use is 5 * 5 *5 for convolution operation, and the step size of convolution operation is 2. As the data progresses through different stages along the compression path, its

resolution is reduced. Because the second operation extracts features by considering only non-overlapping 2×2×2 volumes, the size of the resulting feature map is halved. The operation process is roughly like Figure2: This operation acts somewhat like our pooling layer. (We found that in some image recognition benchmarks, max pooling can simply be replaced by a convolutional layer with increased stride without loss of accuracy), so in this method it is replaced by using a convolutional operation. Replacing pooling operations with convolutions also results in networks that, depending on the implementation, can have a smaller memory footprint during training, since there is no switch to map the output of a pooling layer back to its input needed for backpropagation through only Applying deconvolution instead of unpooling allows for better understanding and analysis.

2.3.3 Residual

The depth of the deep learning network has a great influence on the final classification and recognition effect, so the normal idea is to design the network as deep as possible. But in fact, this is not the case. When the network is very deep, the effect of conventional network stacking (plain network) becomes worse and worse. That is, the deeper the network, the more obvious the phenomenon of gradient disappearance, and the training effect of the network will not be very good. But now the shallow network cannot significantly improve the recognition effect of the network, so the problem to be solved now is how to solve the problem of gradient disappearance in the case of deepening the network.

2.4 Downsampling

Downsampling allows the method to reduce the size of the signal presented as input and increase the receptive field of subsequent nets. Instead, our next stage will double the Feature of the previous stage. Make the image conform to the size of the display area and generate a thumbnail of the corresponding image.

2.5 Upsampling

The right part of the network extracts features and extends the spatial support of lower-resolution feature maps in order to collect and combine the necessary information to output a two-channel volume segmentation. The main purpose of upsampling is to enlarge the original image. The enlargement operation on the image cannot bring more information about the image, so the quality of the image will inevitably be affected. However, some algorithms can be used to increase the information of the image, so that the quality of the zoomed image exceeds that of the original image.

2.6 Innovation of V-Net

The residual is introduced. In each stage, Vnet adopts the short-circuit connection method of ResNet (the residual connection in the horizontal direction uses element-wise); the convolutional layer replaces the pooling layer of upsampling and downsampling. V-Net is a deformation of U-net. The data set at this time can directly use the 3D data set. The final output is also single-channel 3D data. Therefore, what they did in this paper was a two-category problem, so they used the Dice loss commonly used in medical images.

2.7 Skip-Connection

We forward the features extracted from the early stages of the left part of the CNN to the right part. This is represented schematically in Figure 2 by horizontal connections. In this way, we can collect fine-grained details that would be lost in the compression path, and we can improve the quality of the final contour prediction. We also observe that when these connections improve the convergence time of the model. The original intention of Skip connection is to solve the problem of gradient vanished. When learning the parameters of a deep neural network, it is usually through the gradient descent method, that is, the gradient of each layer is calculated from the output layer (output layer) of the network to the input layer (input layer). Since the gradient is usually a value less than 1, when there are many layers, the gradient will become smaller and smaller. Finally, there is the problem of gradient vanish. When the gradient is infinitely close to 0, there is no way for the network to update and learn. In order to solve this problem, there is the idea of skip connection: in short, an additional shallow input is added to the middle layer of the deep network, so that the "path" of the gradient is no longer so long. Similar to providing a compound path, on the basis of the original "long path", now an additional "shortcut" is added. Skip connection is essentially an additional "shortcut" to calculate the gradient.

2.8 Dice Coefficient

In the mask model for predicting brain tumors or skin lesions, we generally classify the primitives in the mask image as 1 or 0, that is, if there is a mask in the primitive, we declare it as 1, and if there is no mask in the primitive, we declare is 0, this binary classification of images on a per-pixel basis is called "semantic segmentation". If we are trying to recognize many objects in an image, it is called "instance segmentation", and instance segmentation is a type of multi-class segmentation. For example, in the field of view of a self-driving car, objects are classified as cars, roads, trees, houses, sky, pedestrians, etc. In semantic (binary) segmentation and instance (multi-class) segmentation, we need a loss function to compute the gradient. The Dice coefficient is a set similarity measurement function, which is usually used to calculate the similarity between two samples, and the value is [0, 1].

3 Method[9][10]

3.1 Introduction

The diagnosis can be distinguished either as benign, or malignant based on the provided feature set and it's a binary classification problem. We can train a model that help us to do the prediction.

3.2 Import Libraries

The first step of training a cancer prediction model is to import libraries, such as "Pandas", a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, and "Numpy" offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more. Besides, there is "matplotlib.pyplot" which is a collection of functions that make matplotlib work like MATLAB, and MATLAB is a programming and numeric computing platform used by millions of engineers and scientists to analyze data, develop algorithms, and create models. We also import "Seaborn" which is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Finally, "Scikit-learn"(sklearn) provides a large amount of common machine learning algorithms and functional dataset.

3.3 Load Dataset

The second step is to load the dataset. Since it's difficult to use real pictures to train models of predicting cancer cells, most of the models use spreadsheets such as ".scv" files that contain characteristics of real pictures of cells. There are several characteristics such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry and fractal dimension. This kind of form containing features of cells can help us finish the step of loading the data.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	id	diagnosis	Radius_m	Texture_m	perimeter_	area_mean	smoothnes	compactne	concavity_	concave p	symmetry	fractal_dir	radius_se	texture_se	perimeter_	area_se	smoothnes	compactne	concavity_	concave p	symmetry	fractal_dir	radius_wo
2	842302	M	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871	1.095	0.9053	8.589	153.4	0.006399	0.04904	0.05373	0.01587	0.03003	0.006193	25.38
3	842517	M	20.57	21.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	0.5435	0.7339	3.398	74.08	0.005225	0.01308	0.0186	0.0134	0.01389	0.003532	24.99
4	84300903	M	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999	0.7456	0.7869	4.585	94.03	0.00615	0.04006	0.03832	0.02058	0.0225	0.004571	23.57
5	84348301	M	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744	0.4956	1.156	3.445	27.23	0.00911	0.07458	0.05661	0.01867	0.05963	0.009208	14.91
6	84358402	M	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809	0.05883	0.7572	0.7813	5.438	94.44	0.01149	0.02461	0.05688	0.01885	0.01756	0.005115	22.54
7	843786	M	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07613	0.3345	0.8902	2.217	27.19	0.00751	0.03345	0.03672	0.01137	0.02165	0.005082	15.47
8	844359	M	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	0.1794	0.05742	0.4467	0.7732	3.18	53.91	0.004314	0.01382	0.02254	0.01039	0.01369	0.002179	22.88
9	84458202	M	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	0.2196	0.07451	0.5835	1.377	3.856	50.96	0.008805	0.03029	0.02488	0.01448	0.01486	0.005412	17.06
10	844981	M	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235	0.07389	0.3063	1.002	2.406	24.32	0.005731	0.03502	0.03553	0.01226	0.02143	0.003749	15.49
11	84501001	M	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	0.203	0.08243	0.2976	1.599	2.039	23.94	0.007149	0.07217	0.07743	0.01432	0.01789	0.01008	15.09
12	845636	M	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323	0.1528	0.05697	0.3795	1.187	2.466	40.51	0.004029	0.009269	0.01101	0.007591	0.0146	0.003042	19.19
13	84610002	M	15.78	17.89	103.6	781	0.0971	0.1292	0.09954	0.06606	0.1842	0.06082	0.5058	0.9849	3.564	54.16	0.005771	0.04061	0.02791	0.01282	0.02008	0.004144	20.42
14	846226	M	19.17	24.8	132.4	1123	0.0974	0.2458	0.2065	0.1118	0.2397	0.078	0.9555	3.568	11.07	116.2	0.003139	0.08297	0.0889	0.0409	0.04484	0.01284	20.96
15	846381	M	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364	0.1847	0.05338	0.4033	1.078	2.903	36.58	0.009769	0.03126	0.05051	0.01992	0.02981	0.003002	16.84
16	84667401	M	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025	0.2069	0.07682	0.2121	1.169	2.061	19.21	0.006429	0.05936	0.05501	0.01628	0.01961	0.008093	15.03
17	847990002	M	14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364	0.2303	0.07077	0.37	1.033	2.879	32.55	0.005607	0.0424	0.04741	0.0109	0.01857	0.005466	17.46
18	848406	M	14.68	20.13	94.74	684.5	0.09867	0.072	0.07395	0.05259	0.1586	0.05922	0.4727	1.24	3.195	45.4	0.005718	0.01162	0.01998	0.01109	0.0141	0.002085	19.07
19	84862001	M	16.13	20.68	108.1	798.8	0.117	0.2022	0.1722	0.1028	0.2164	0.07356	0.5692	1.073	3.854	54.18	0.007026	0.02501	0.03188	0.01297	0.01689	0.004142	20.96
20	849014	M	19.81	22.15	130	1260	0.09831	0.1027	0.1479	0.09498	0.1582	0.05395	0.7582	1.017	5.865	112.4	0.006494	0.01893	0.03391	0.01521	0.01356	0.001997	27.32
21	8510426	B	13.54	14.36	87.46	566.3	0.09779	0.08129	0.06664	0.04781	0.1885	0.05766	0.2699	0.7886	2.058	23.56	0.008462	0.0146	0.02387	0.01315	0.0198	0.0023	15.11
22	8510653	B	13.08	15.71	85.63	520	0.1075	0.127	0.04568	0.0311	0.1967	0.06811	0.1852	0.7477	1.383	14.67	0.004097	0.01898	0.01698	0.00649	0.01678	0.002425	14.5
23	8510824	B	9.504	12.44	60.34	273.9	0.1024	0.06492	0.02956	0.02076	0.1815	0.06905	0.2773	0.9768	1.909	15.7	0.009606	0.01432	0.01985	0.01421	0.02027	0.002968	10.23
24	8511133	M	15.34	14.26	102.5	704.4	0.1073	0.2135	0.2077	0.09756	0.2521	0.07032	0.4389	0.7096	3.384	44.91	0.006789	0.05328	0.06446	0.02252	0.02672	0.004394	18.07

Fig. 5 This is the form of the data we use when training the model.

3.4 Analyze Data

After the dataset is read, we can analyze our data. We use some functions to show the number of elements in each dimension, and it's also possible to see some basic statistical details, such as percentage, mean, standard deviation, etc. Moreover, we can count the amount of benign and malignant cells, and draw a bar graph to show the result. We can tell whether the dataset is balanced by the graph. Then, there is an important step which is to drop specific row or column to ignore the data that are not necessary now. After dropping them, we can draw line graph to check the distribution of the numerical feature, and also if they are skewed or if they follow the normal distribution. Last, we draw heat map which is commonly used in machine learning to discriminate the correlation between features. Functions are used for generating a table-like layout for component placement and a colorbar to aid the chart. These steps help us know the characteristic of the dataset better and how the data distribute.

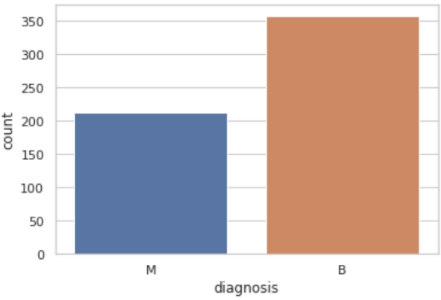

Fig. 6 This is the bar graph after analyzing the data, and it shows that dataset is balanced.

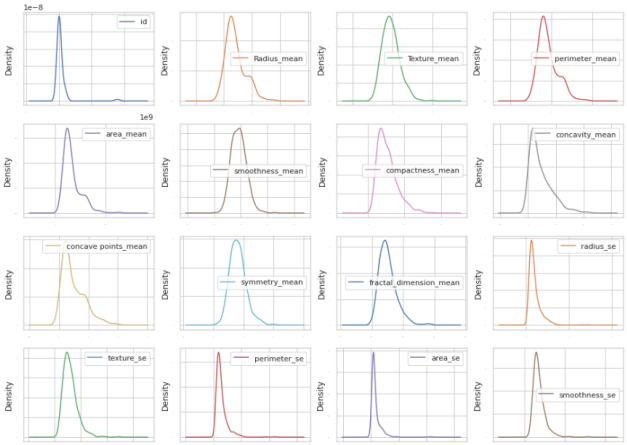

Fig. 7 This is the line graph after analyzing the data, and it shows that even though some features left skewed, all of them follows the normal distribution.

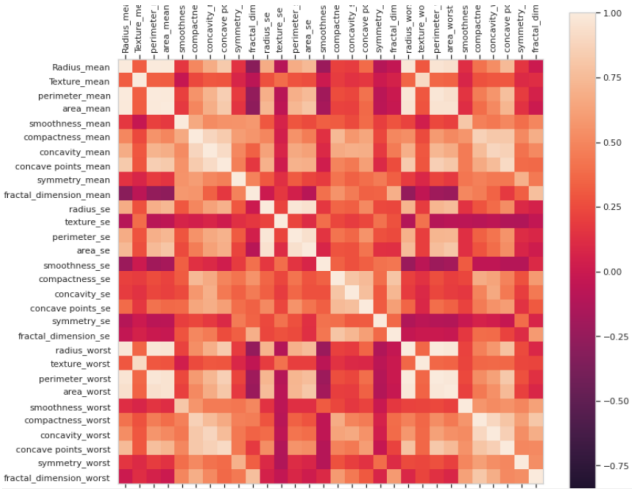

Fig. 8 This is the heat map after analyzing the data, and it shows that there are a lot of features that have strong correlation to each other.

3.5 Prepare Data and Evaluate Algorithms

Then, we prepare data for the model. There are things we have to do such as drop the duplicate rows, split the training data, and scale data. We evaluate the accuracy of different algorithms, and we can observe that scaling the data can improve the performance of some algorithms dramatically. Besides, we choose the best algorithm for us to train the model.

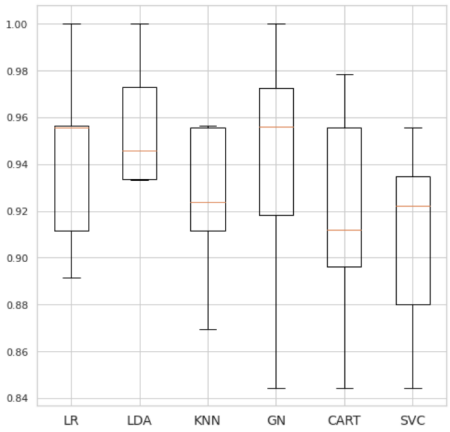

Fig. 9 This is the figure of algorithm comparison.We can see that the best accuracy score is provided by LDA with 96% and the Decision Tree follows it with %91.

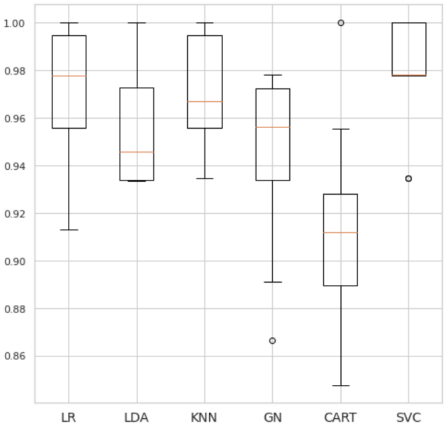

Fig. 10 This is the figure of algorithm comparison after scaling the data. It's obvious that scaling the data improved the performance of the Support Vector Classification dramatically.

3.6 Finalize the Model

After things are all prepared, we can finally finalize the model. Hyper-parameters are parameters that given in advance based on experience before model training, and the trained models will be different based on different hyper-parameters. Though there are just slight differences between them, the step of hyper-parameter tuning can choose the best parameters and thus improve the performance of the model. Moreover, K-Fold Cross-Validation help us to avoid bias depending on a specific training and testing data. In this method, we split data into k consecutive folds, and each fold is then used once as a validation while the k - 1 remaining folds form the training set. In this way, k models are trained, and the average of k modeling results is used as the final result of this hyper-parameter combination, so as to obtain the optimal solution of the model.

3.7 Test the Model

The last step is to test the model. We have to train the model to test it first, and then we calculate the accuracy score of the prediction. It's also good to draw a confusion matrix to show the result of prediction. Confusion matrix is a specific table layout that allows visualization of the performance of an algorithm in the field of statistical classification problem and machine learning, especially in supervised learning.

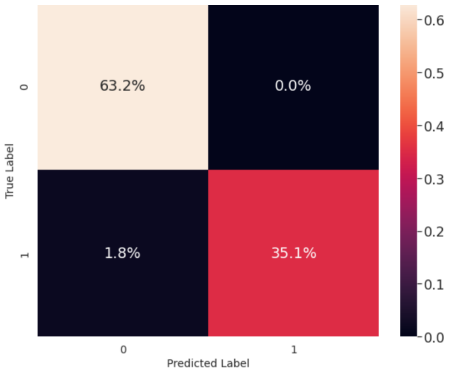

Fig. 11 This is the confusion matrix of the result. We can see that the percentage of True Positive is 35.1%, False Negative is 1.8%, True Negative is 63.2%, and False Positive is 0.0%. It's not perfect but it already has low probability to have error, and the accuracy rate is approximately 98%.

4 Results

There are some comparison that we can do after finishing the model. For instance, the SVC provided the best accuracy score with the given hyperparamters. Moreover, we also observed that scaling the data improved the performance of the algorithm for the sample dataset. To make the model better, outliers can be removed to check if improves the performance or not as the next step.

5 Problems and Prospect

5.1 Problems of Image Recognition

It is difficult to predict exactly what the future of image recognition in medicine will look like, as it depends on a variety of factors such as technological advancements and data deficiencies. However, it is likely that image recognition in medicine will continue to make significant progress in the coming years.

5.2 Prospect of Image Recognition

One area where image recognition is expected to make significant progress is in the development of machine learning algorithms that are able to analyze and interpret medical images with increasing accuracy. These algorithms will likely be trained on larger and more diverse datasets, which will allow them to learn more about the various features and abnormalities that may be present in medical images. This will enable the algorithms to make more accurate and reliable diagnoses, and may also allow them to identify patterns or trends that may not be apparent to human analysts.

Another area where progress is expected is in the integration of image recognition technology with other healthcare technologies. For example, image recognition algorithms may be used in conjunction with electronic health records or other data sources to provide a more comprehensive view of a patient's health. This could allow for more personalized and targeted treatment plans, and may also enable earlier detection of potential health issues.

Overall, it is likely that image recognition in medicine will continue to evolve and improve in the coming years, and will play an increasingly important role in the practice of healthcare.

6 Conclusion

We studied on the image recognition in medicine and tried to implement. In study part, we pick two networks which is frequently used in medical image recognition. The first one is U-Net. It is used in 2D image segmentation. The second one is V-Net. It is a network based on U-Net. It is used in 3D image segmentation. They are all end-to-end convolutional neural networks.

In implement part, we found that there are many libraries and functions to use, so it takes us a lot of time to study on how to utilize them. We choose the dataset "Breast Cancer Prediction" from kaggle. Then, we tried to find the best algorithm to use via six algorithms, LR, LDA, KNN, GN, CART, and SVC. Among them, the SVC algorithm is the best one. After choosing the algorithm, we adjust the hyper-paremeters of model and find the best one. Finally, we train the model and use confusion matrix to see the accuracy score of the prediction.

Although there are many applications of AI in the medical field, there are also some challenges that need to be overcome. For example, in terms of accuracy, AI may not be able to achieve the same accuracy as human doctors, so it is necessary to strengthen the accuracy of AI. In addition, the application of AI in the medical field may also exist some medical problems, such as private protection, the quality of the data.

In conclusion, AI has many advantages in the medical field, but it still needs to be developed and improved to achieve its maximum effectiveness.

References

[1] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation, 234–241 (2015)

[2] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[3] Li, X., Chen, H., Qi, X., Dou, Q., Fu, C.-W., Heng, P.-A.: H-denseunet: Hybrid densely connected unet for liver and tumor segmentation from ct volumes. IEEE Transactions on Medical Imaging **37**(12), 2663–2674 (2018). https://doi.org/10.1109/TMI.2018.2845918

[4] Noori, M., Bahri, A., Mohammadi, K.: Attention-guided version of 2d unet for automatic brain tumor segmentation. In: 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE), pp. 269–275 (2019). https://doi.org/10.1109/ICCKE48569.2019.8964956

[5] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)

[6] Fausto, N.N. Milletari, Ahmadi, S.-A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation (2016)

[7] Laura, M.B., Veronica, V.: Brain tumor segmentation using 3d-cnns with uncertainty estimation (2020)

[8] Du, R., Cao, P., Han, L., Ai, Q., King, A.D., Vardhanabhuti, V.: Deep convolution neural network model for automatic risk assessment of patients with non-metastatic nasopharyngeal carcinoma. arXiv preprint arXiv:1907.11861 (2019)

[9] Mustafa Cakir: Breast Cancer Prediction - 98% Accuracy. https://www.kaggle.com/code/mustafacakir/breast-cancer-prediction-98-accuracy/notebook, Last accessed on 2022/12/27 (2022)

[10] VIVEK TYAGI: Lung Cancer Detection - 95% Positive Cases. https://www.kaggle.com/code/vivtyagi/lung-cancer-detection-95-positive-cases, Last accessed on 2022/12/28 (2022)