© Universiti Tun Hussein Onn Malaysia Publisher's Office

IJIE

http://penerbit.uthm.edu.my/ojs/index.php/ijie
ISSN : 2229-838X e-ISSN : 2600-7916

The International Journal of Integrated Engineering

Subcutaneous Vein Recognition System Using Deep Learning for Intravenous (IV) Access Procedure

Chan Xiao Jing[1], Goh Chuan Meng[1]*, Meei Tyng Chai[1], Sayed Ahmad Zikri Sayed Aluwee[1], Syed Ayaz Ali Shah[2]

[1]Universiti Tunku Abdul Rahman,
Jalan Universiti Bandar Barat, Kampar, 31900, MALAYSIA

[2]COMSATS University Islamabad, Abbottabad Campus,
Abbottabad, KPK, PAKISTAN

*Corresponding Author

DOI: https://doi.org/10.30880/ijie.2023.15.03.007
Received 31 October 2022; Accepted 29 December 2022; Available online 31 July 2023

Abstract: Intravenous (IV) access is an important daily clinical procedure that delivers fluids or medication into a patient's vein. However, IV insertion is very challenging where clinicians are suffering in locating the subcutaneous vein due to patients' physiological factors such as hairy forearm and thick dermis fat, and also medical staff's level of fatigue. To resolve this issue, researchers have proposed autonomous machines to be used for IV access, but such equipment are lacking capability in detecting the vein accurately. Therefore, this project proposes an automatic vein detection algorithm using deep learning for IV access purpose. U-Net, a fully connected network (FCN) architecture is employed in this project due to its capability in detecting the near-infrared (NIR) subcutaneous vein. Data augmentation is applied to increase the dataset size and reduce the bias from overfitting. The original U-Net architecture is optimized by replacing up-sampling with transpose convolution as well as the additional implementation of batch normalization besides reducing the number of layers to diminish the risk of overfitting. After fine-tuning and retraining the hypermodel, an unsupervised dataset is used to evaluate the hypermodel by selecting 10 checkpoints for each forearm image and comparing the checkpoints on predicted outputs to determine true positive vein pixels. The proposed lightweight U-Net has achieved slightly lower accuracy (0.8871) than the original U-Net architecture. Even so, the sensitivity, specificity, and precision are greatly improved by achieving 0.7806, 0.9935, and 0.9918 respectively. This result indicates that the proposed algorithm can be applied into the venipuncture machine to accurately locate the subcutaneous vein for intravenous (IV) procedures.

Keywords: Intravenous (IV) access, subcutaneous vein detection, deep learning, U-Net, unsupervised learning

1. Introduction

Intravenous (IV) access is one of the fundamental medical procedures that delivers fluids or medication into the vein such as blood withdrawal, drug insertion or transfusion. Approximately 80 percent of patients required IV access at some stage during their hospitalization [1]. However, IV insertion is not a straightforward procedure where clinicians are suffering in locating the subcutaneous vein due to many physiological factors such as hairy forearm, dark skin tone, obese patients, elderly patients, and a high volume of dermis fat [2]. This limitation has caused the patients to suffer from venipuncture pain and get injured due to multiple attempts of IV insertion. In addition, medical staff will be difficult to identify the exact vein location on patients' arms in a fatigue condition, which causes a higher possibility for the patients to experience pain, injury, or worse, venous thrombosis.

**Corresponding author: gohcm@utar.edu.my*

2023 UTHM Publisher. All rights reserved.

Among the methods developed to secure IV access safely, NIR is the optimum imaging technology for subcutaneous vein localization to improve IV success rate. The deoxygenated haemoglobin in the vein vessels can absorb the infrared light when the light passes through the hand or arm location or reflects by that location [3]. With the aids of infrared cameras and other related devices, the vein patterns will appear as dark lines. These lines provide a "map" that can be used to guide medical staff for efficient IV insertion [2]. Despite the effectiveness of NIR subcutaneous vein localization in initial stages, later studies reported its limitation in detecting the vein accurately, included first-attempt IV insertion success rates, total procedure time, and lack of availability due to expensive cost [2], leaving the problem unsolved yet till date. Therefore, the use of deep learning techniques is essential to improve subcutaneous vein localization when using NIR imaging devices and also assist medical staff to carry out IV insertion more accurately.

This paper presents an improved deep learning-based subcutaneous vein extraction technique by proposing a fully connected convolutional network model called U-Net originally developed in [4]. This optimised lightweight U-Net model is aimed to be implemented in intelligent healthcare machines to replace manual IV insertion. As CNN and transfer learning as seen in [3] and [5] are typically used for biomedical data classification, the U-Net architecture outperforms these deep learning techniques in biomedical image processing, in which the desired biomedical features such as brain tumour and blood vessels can be localized, and their features can be extracted more efficiently. The experiment of this project utilized data augmentation to generate a larger batch of image data before feeding the image data into the lightweight U-Net model [4]. The original vein images can be augmented in various transformations including rotation, translation, zooming, and shifting to create a diverse modified training set as well as to expand the size of the training set. This allows the model to learn invariance from the diverse deformations efficiently besides resolving the limitations of lacking training data images. Unsupervised learning of vein image segmentation is explored by comparing 10 checkpoints on forearm images and predicted outputs to determine the number of true positive pixels that can be extracted by the proposed model.

2. Related Works

Literature review has been done on several deep learning approaches including CNN, DenseNet, AlexNet, VGG-Net, ResNet, Inception-ResNet, and U-Net. Among these reviewed techniques, CNN is the most common network architecture in vein segmentation research as seen in [3], [5], and [6]. Transfer learning using DenseNet-161 is applied in [7] and [8] and has achieved higher performance than CNN overall. However, AlexNet, which is another pretrained model used in [9] and [10] achieved a lower performance compared to DenseNet-161 because it is shallower. Besides, VGG-Net in [11], [12] and [13], ResNet in [14] and [15], and Inception-ResNet in [16] have achieved desirable results from the segmentation, but ResNet and Inception-ResNet can incur long training time due to deep architecture. Furthermore, VGG-Net applied in [11] has performed poorly in poor quality images, resulting in low EER.

The last reviewed network architecture is U-Net in [17], [18], [19], and [21], which has been widely used in many biomedical tasks and it is the most accurate deep learning (DL) model till-date. U-Net, originally developed in [4], is a modified CNN architecture favoured by biomedical image processing due to its robustness in biomedical image segmentation. U-Net consists of encoder layers which reduce image dimensions, and decoder layers that repair the detailed features of spatial dimension during up-sampling. The bottleneck layer mediates between the encoder layers and decoder layers.

In [17], a Gabor filter was used in the first encoding block of a modified U-Net architecture. The ROIs were first thresholded to a foreground value to label vein and background pixels. Then, morphological operations which were dilation and erosion were applied to remove noises from the images. An end-to-end U-Net model in [18] was first used to segmentate the vein images using four baseline algorithms and the resulting image is binarized by a threshold of 0.5 to label the vein pixels. The binarized images were used for model training to get the probability score of each pixel and to generate predicted output through thresholding. The extraction process was fast and simple with a moderate network size yet ensured satisfied outputs. However, vein pixel labelling was relied on multiple handcrafted algorithms which might not be effective in the long term.

A deformable convolutional network by modifying the standard U-Net architecture [19] can capture the complex venous structural features effectively. The convolutional layers in standard U-Net were replaced by deformable layers to adjust receptive fields adaptively and residual recurrent layers for effective depth mining and feature accumulation. This modified architecture was able to obtain a low EER and high extraction ability as compared to the original U-Net.
Latest research on dorsal hand vein segmentation reported that a best modified UNet from the other nine different U-net networks, known as ResNet-UNet based CNN method, is able to achieve Dice scores at 95.11 for the vein segmentation [20]. This statement reveals that U-Net based deep learning algorithm is an outstanding method in segmentation the vein. However, the integration of U-Net, RNN, ResNet and deformable neural networks might be complicated and heavy duty to be trained and thus might not be able to implement to venipuncture system in detecting NIR vein.

Another finger vein recognition system had compared between three semantic deep learning models, which were U-Net, RefineNet and SegNet [21]. U-Net was trained on an SGD optimizer with learning rate of 0.08 for 300 epochs. Similarly, SegNet was trained SGD optimizer with a smaller learning rate of 0.003 for 30,000 epochs. RefineNet was trained on Adam optimizer with a much smaller learning rate of 0.0001 for 40,000 epochs. Each network used different sets of training parameters which might lead to biased results, but it was notable that U-Net obtained the lowest EER in overall and outperformed maximum curvature extraction technique.

Among these techniques, U-Net achieves an outstanding performance due to its capability and favourability towards biomedical image segmentation. However, most of the previous U-Net developments focus on finger vein or palm vein recognition for biometric verification purpose but no specific study was reported on Near Infrared (NIR) subcutaneous vein extraction. Therefore, this paper proposed an improved U-Net model with simplified system architecture for NIR subcutaneous vein extraction that tend to be used in automatic venipuncture machine for IV insertion issue. One should be noted that the vein detection algorithm should be as simplified as possible in order to implement into the real-world venipuncture machine.

3. Material and Method

3.1 Dataset

An in-house vein dataset that contains 8 sets of forearm images and their respective groundtruth collected by Universiti Tunku Abdul Rahman is used to carry out intensive model training. Since it is a small dataset, data augmentation is applied in each training to generate more than 20,000 augmented images to be inputted into the model in a real time manner. Fig. 1 and Fig. 2 show the original image and some examples of augmented images respectively.

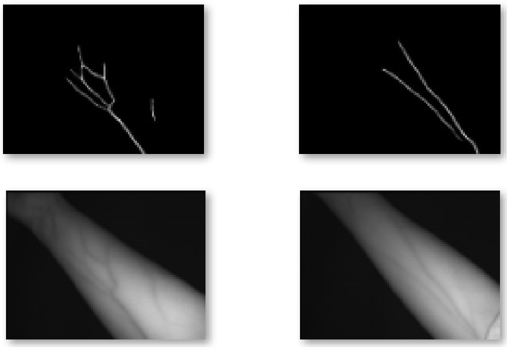


Fig. 1 - Examples of forearm images and their respective groundtruth

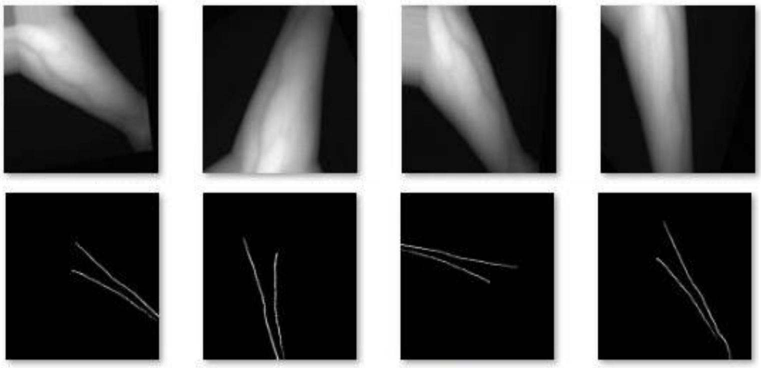


Fig. 2 - Examples of augmented images

3.2 Intensive Model Training

The training process is divided into three parts: 1) optimize model architecture, 2) data augmentation, and 3) hyperparameters utilization. All the training generators contain training and validation subsets with a validation split of 0.3. Each data generator was inserted to the training process for real time augmentation during the model training to train the different batches of augmented images in each epoch.

3.2.1 Initial U-Net Architecture

During the first stage of model training, the original U-Net [17] is modified to specifically fit with the detection of NIR vein features. For the expansion blocks of U-Net, instead of using up sampling layers as in the original architecture, 3×3 transpose convolution layers (Conv2DTranspose) with stride of 2 are used to carry out convolution in an opposite direction using some parameters rather than simply scaling up the images. The output of transpose convolution can be computed as follows:

$$o = (i-1) * s + k - 2p \quad (1)$$

where o denotes output activation size of a transpose layer, i denotes the input dimension, s denotes stride, k denotes filter size, and p denotes as padding.

Besides, batch normalization layers are added in the contracting path. The reason for adding batch normalization layers between the convolutional layers and ReLU activation layers is to standardize the output of a layer before they pass to the next block so that the covariance shift problem can be minimized and make the network training more stabilized. The output layer is a 1×1 convolution layer with the application of sigmoid activation instead of SoftMax activation in the original U-Net architecture. Another model optimization done is the use of Adam optimizer instead of SGD optimizer in the original U-Net model as Adam optimizer is more robust to handle sparse gradients along with the computation of individual learning rates for different network parameters. The initial modified U-Net architecture at this stage is shown in Fig. 3.

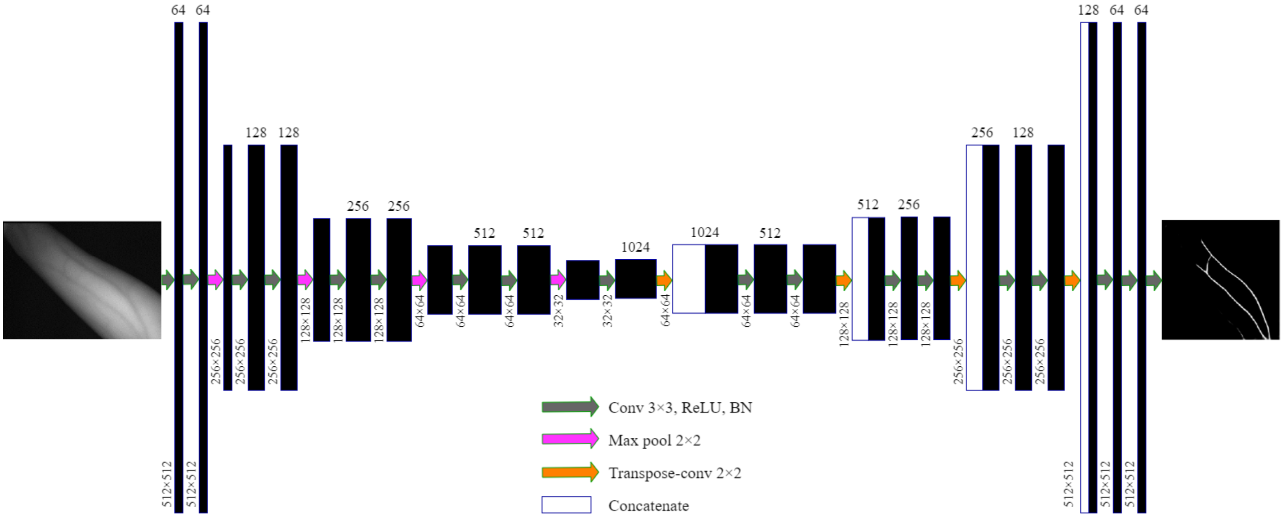


Fig. 3 - Initial modified U-Net architecture

3.2.2 Data Augmentation

Data augmentation was carried out to increase the amount of training data since the dataset obtained was small as well as to avoid overfitting. The ImageDataGenerator class from Keras library was implemented to achieve data augmentation with the augmentation techniques. Four combinations of data augmentation techniques were used to conduct experiments to determine the most suitable combination of data transformation techniques. Table 2 showed all the tested combinations techniques.

The pre-processing of the image augmentation was setup into three phases prior to image training. Firstly, each augmented image, including the forearm images and groundtruth images, is resized into 256×256 pixels. Secondly, the images are normalized by dividing the pixels with 255 so that the pixels ranged between 0 and 1. Thirdly, groundtruth images are one-hot encoded using a predefined threshold to assign vein pixels with “1” and non-vein pixels with “0”.

3.2.3 Hyperparameter Fine-Tuning

The initial modified model is furthered trained on different hyperparameter settings including learning rates (0.001, 0.0001, 0.00001), activation functions (ReLU, ELU), and number of epochs (5, 10, 15, 25) in order to optimize the best model performance. The best configuration of all these settings is determined to be used to calculate performance metrics and subsequent hyperparameter fine-tuning.

3.3 Lightweight U-Net Architecture

After intensive model training (Section 3.2), the initial modified U-Net model (as shown in Fig. 3) is further simplified before hyperparameter fine-tuning by reducing the number of layers. The purpose of this optimization is to reduce the risk of overfitting the training data. The complexity of the U-Net model was reduced by having the highest filter size in the bottleneck layers as 512 instead of 1024 to trim down several hidden layers. The network architecture of the proposed lightweight U-Net is shown in Fig. 4. Subsequently, the model is used for hyperparameter fine-tuning to identify the optimum learning rate and retrain the model.

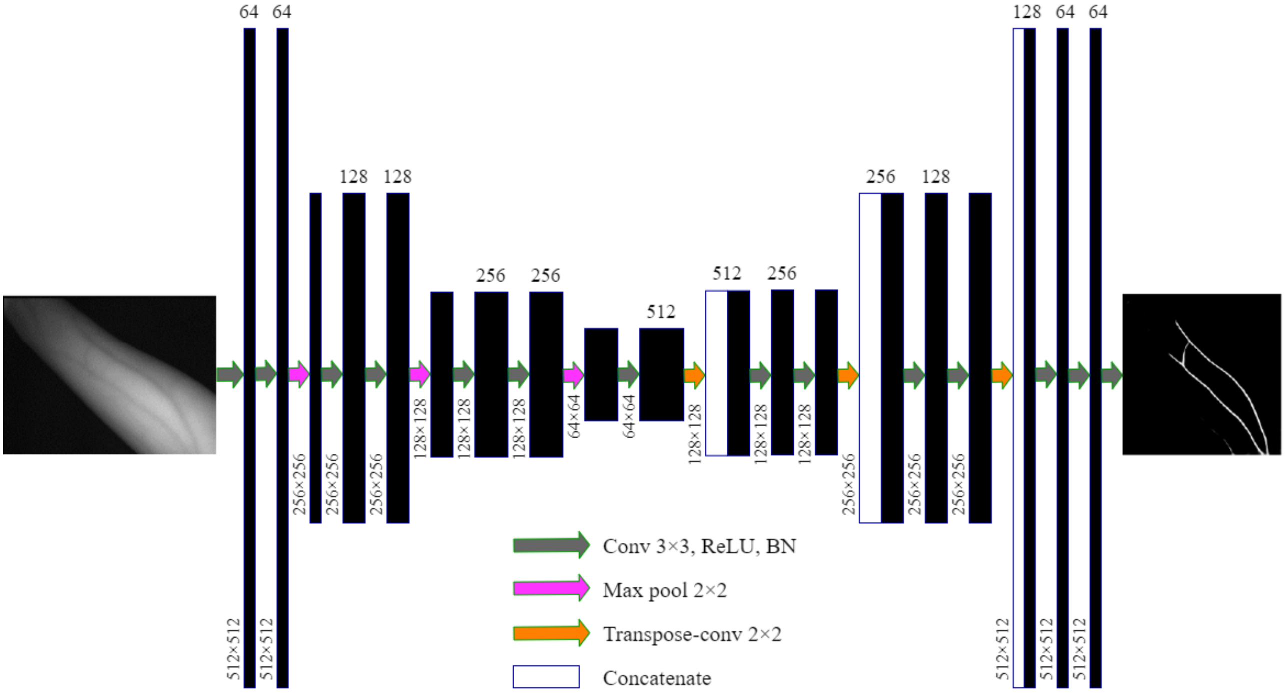


Fig. 4 - Lightweight U-Net architecture

3.4 Unsupervised Vein Segmentation

Since the model works well on the supervised dataset, we explore its capability to extract vein pixels from an unsupervised dataset collected at Universiti Tunku Abdul Rahman in which it contains only 31 subject forearm images without groundtruth. The unsupervised dataset is more complicated than the original dataset as there is a larger difference and more noises contained in the forearm images, such as wristband, thick dermis fur, and darker skin tone. Fig. 5 shows some of the forearm images in the dataset.

The retrained hypermodel is used to predict and evaluate the performance of the proposed model in detecting vein for real world application with this unsupervised dataset. Prior the model evaluation, we have selected 20 checkpoints per forearm image in which the first 10 checkpoints are positive vein pixels while another 10 checkpoints are negative vein pixels such as background pixels or wristbands. The checkpoints were chosen randomly from raw images, and it is used to compare with the corresponding pixel in predicted output image (Refer to Fig. 7 (b)) to check the reliability of the proposed model in detecting vein without bias where the comparison should achieve zero deviation. The checkpoints are referring to the true vein pixels and false vein pixels from raw images (unsupervised dataset) as shown in Fig. 5, where true vein pixels referring to the vein pixel while false vein pixels referring to non-vein pixel. Therefore, the checkpoints are chosen from random direction including the main branch of the vein and also fork vein branches to ensure that the result obtained is reliable. From the 10 positive vein pixels, we can calculate true positives (TP) and false negatives (FN) to find out the how good the model can perform to extract the veins. While from the 10 negative pixels, true negatives (TN) and false positives (FP) can be calculated to determine model performance on identify nearby machines or noises. Lastly, these four measurements were used to compute the performance metrics including dice coefficient, accuracy, specificity, sensitivity, and precision. The equations to compute the performance metrics are shown in section 4.3, Eqn. 2 to Eqn. 6.

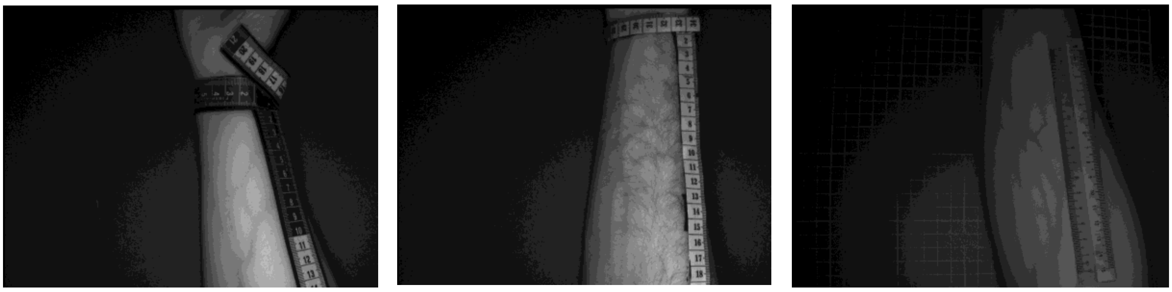


Fig. 5 - Examples of forearm images in the unsupervised dataset

4. Experiments and Analysis

4.1 Analysis on Initial Model Training

Model training starts with the optimized U-Net that used Conv2DTranspose layers instead of original UpSampling layers in the expansion path. The predicted vein image of the first optimized U-Net architecture contains noise in which the forearm textures are wrongly extracted as vein features. The low dice coefficient score in Table I (model version 1) indicates that this optimized model is less powerful to identify positive pixels correctly, resulting in a smaller overlap of vein pixels between predicted vein image and groundtruth.

After that, the model is further optimized by adding batch normalization layers between convolution layers and activation layers to standardize the input before entering subsequent convolutions. Batch normalization can maintain the input with variance of 1 and mean of 0, resulting in less covariance shift issue and reduced the sensitivity of the current layer towards the variations in input distribution. The dice coefficient score obtained is higher than the previous optimized network architecture. Less noise is present in the predicted vein image, which indicates that the vein pixels can be detected precisely as compared to the first U-Net model.

Table 1 - Comparison between optimized network architecture

Model Version	Comparison Aspects	Dice Coefficient
1	Replaced UpSampling layers with Conv2DTranspose layers.	0.6239
2	Added Conv2DTranspose layers and BatchNormalization layers.	0.7032

Next, the model is trained with different augmentation techniques to obtain the best data augmentation combination that can extract vein features with the least noise while at the same time, retain most of the useful vein features. It is decided that the first combination of data augmentation techniques would be used for subsequent simulations because it has the most satisfied predicted output in terms of dice coefficient and groundtruth with the least presence of noise and loss of vein features. The results of all data augmentation combinations are shown in Table 2 below.

Table 2 - Comparison between data augmentation techniques

Combination	Comparison Aspects	Dice Coefficient

1	rotation_range = 360,
	width_shift_range = 0.05,
	height_shift_range = 0.05,
	shear_range = 0.3,
	zoom_range = 0.3,
	horizontal_flip = True,
	vertical_flip = True,
	fill_mode = ‘nearest’
		0.7032

2	zoom_range = 0.3,
	shear_range = 0.3,
	fill_mode = ‘constant’,
	cval = 0.,
	horizontal_flip = True,
	vertical_flip = True
		0.7895

3	brightness_range = [-1, 1],
	rotation_range = 360,
	zoom_range = [0.5, 1.5],
	horizontal_flip = True,
	vertical_flip = True,
	fill_mode = ‘constant’
		0.3671

4	brightness_range = [-1.5, 1.5],
	shear_range = 0.5,
	fill_mode = ‘nearest’,
	rotation_range = 360,
	width_shift_range = 0.2,
	height_shift_range = 0.2
		0.3307

Then, the model is trained with different hyperparameters including learning rate, activation function, and number of epochs. The model training that uses a learning rate of 0.0001 achieved the highest dice coefficient score and clearest predicted groundtruth among all the learning rates. The model with the largest learning rate (0.001) performs slightly better than the model with the smallest learning rate (0.00001), but both the learning rates have extracted unnecessary forearm features as shown at upper left corners of the sample output images. These extracted noises might increase complexity to localize the veins precisely. Since the learning rate of 0.001 has a higher dice coefficient than learning rate of 0.00001, it can be assumed that the optimal learning rate lies between the range of 0.0001 to 0.001. This range will be used in subsequent research for a proper learning rate fine-tuning.

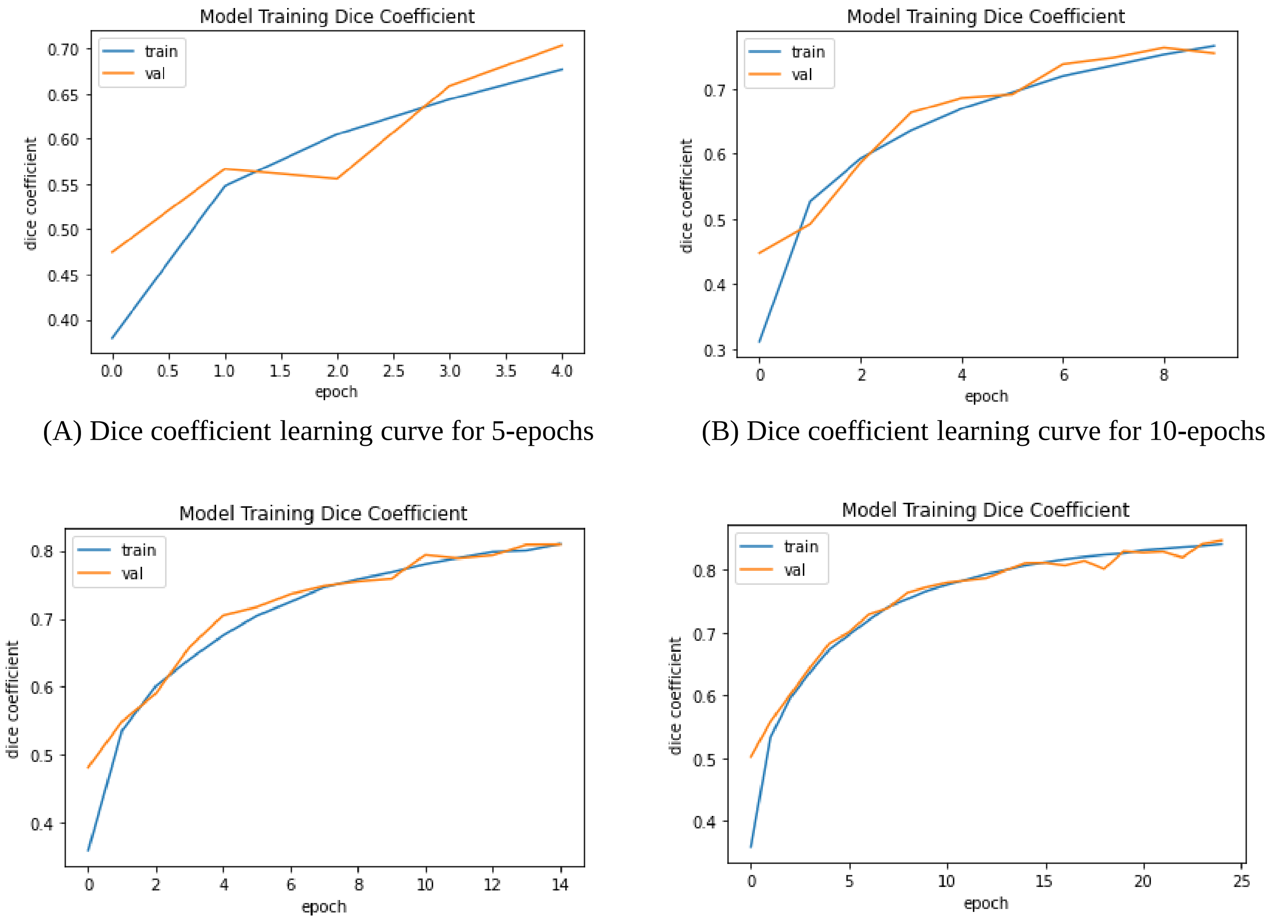
For activation function, both ReLU and ELU activations reflect nearly similar amounts of feature information loss from groundtruth. From a theoretical perspective, ReLU is favoured by most of the deep learning network because it does not saturate in the positive region, resulting in less vanishing gradient effect. However, ReLU is still saturated in the negative region. This might cause a large number of neurons to be inactive if most of the weighted inputs are negative. As a consequence, the activation output is zero and network parameters are unable to be updated. ELU resolves this issue by having negative saturation regions which enable neurons to produce negative activation outputs instead of zero activation. Dead ReLU problem can be avoided and the negative activation outputs are able to update network parameters in correct directions.

However, there is a lack of state-of-the-art models that implement ELU activation in their deep neural network architectures. For example, U-Net architecture developed by [17] and [18] had just followed the existing implementation of ReLU activation in their U-Net models because ReLU is more well-established and more suitable to be used than other activations in the hidden layers. Since the results of ReLU activation shows a higher dice coefficient and there is not much research that supports the implementation of ELU activation, ReLU activation is more suitable to be applied in this vein extraction U-Net model.

For number of epochs, the vein features learned by the model become more complete and clearer when the number of epochs used increases. However, a model will face the risk of overfitting when it is trained with too many epochs [23]. From the learning curves shown in Fig. 6, it is assumed that the models had not overfitted the images since the training and validation dice coefficients were close to each other, but this statement need to be further validated by independent testing dataset. In this training section, since 25 epochs gives the most satisfied output, it is chosen to be applied in the model.

Table 3 - Comparison between hyperparameters

Model Version	Comparison Aspects	Dice Coefficient
1	learning rate = 0.001	0.6498
2	**learning rate = 0.0001**	**0.7032**
3	learning rate = 0.00001	0.5330
4	**activation = ‘relu’**	**0.7032**
5	activation = ‘elu’	0.6529
6	epochs = 5	0.7032
7	epochs = 10	0.7541
8	epochs = 15	0.8092
9	**epochs = 25**	**0.8476**



(A) Dice coefficient learning curve for 5-epochs

(B) Dice coefficient learning curve for 10-epochs

(C) Dice coefficient learning curve for 15-epochs

(D) Dice coefficient learning curve for 25-epochs

Fig. 6 - Dice coefficient learning curve when different numbers of epochs were used

4.2 Analysis on the Refined Lightweight Model

A lightweight U-Net with less convolution layers is defined and fine-tuned to determine the possibility that the original U-Net model might overfit the data, and to determine the performance of a smaller and less complex model in extracting the vein pixels. From the results shown in Table 4, the lightweight U-Net has obtained a higher dice coefficient, indicates that the simpler network in addition of fine-tuned hyperparameter (learning rate) is more robust in extracting vein pixels.

Table 4 - Results obtained by original U-Net and lightweight U-Net

Model Version	Comparison Aspects	Dice Coefficient
1	Original U-Net (Added Conv2DTranspose layers and BatchNormalization layers)	0.8476
2	**Lightweight U-Net after hyperparameter fine-tuning**	**0.8513**

4.3 Evaluation on Model Prediction Using Unsupervised Dataset

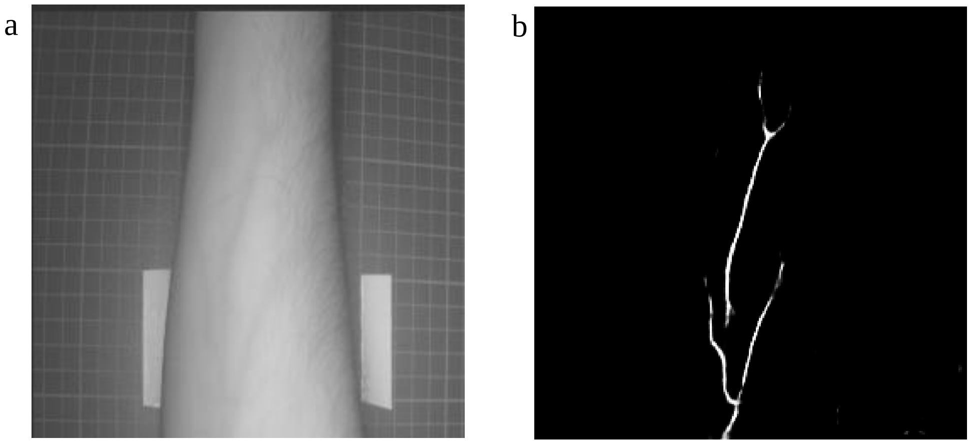


Fig. 7 - Example of good vein extraction result (a) forearm image; (b) predicted output

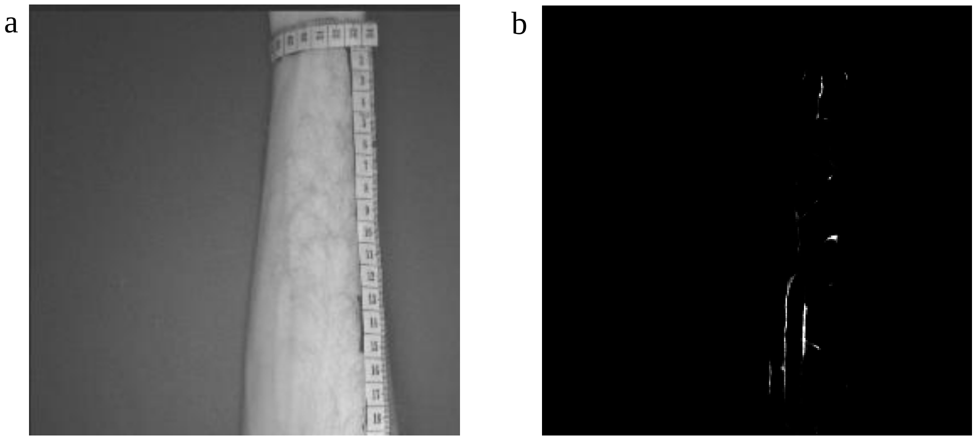


Fig. 8 - Example of bad vein extraction result (a) forearm image; (b) predicted output

Fig. 7 above shows an example of model prediction result, in which this example shows a quite satisfying and complete vein extraction. However, there are also a few outputs that show a poor extraction such as the example shown in Fig. 8. This might be due to the existence of noise in the images such as forearm hair, dermis fat, and dark skin tone that increase the complexity for the model to identify the true vein pixels. As can be seen in Fig. 8, the model has extracted some forearm hair pixels as vein pixels, indicates that post-processing with length filter is needed to have a higher specificity in recognizing the true veins from the image. The artifact such as hair can be further filtered by the length and width of the extracted pixels because the true vein has a unique continuous tubular shape as compared to other artifact's features.

Table 5 - Results obtained for model prediction on unsupervised dataset

Performance Metric	**Result**
Dice coefficient	0.8736
Global accuracy	0.8871
Specificity	0.9935
Sensitivity	0.7806
Precision	0.9918

The result of model prediction on unsupervised dataset is shown in Table 5 above. There are 5 performance metrics used to evaluate unsupervised vein segmentation performance, which are dice coefficient, accuracy, specificity, sensitivity, and precision. The equations are listed as below:

$$Dice\ coefficient = \frac{2TP}{2TP + FP + FN} \tag{2}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3}$$

$$Specificity = \frac{TN}{TN + FP} \tag{4}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{5}$$

$$\mathrm{Pr}\,ecision = \frac{TP}{TP + FP} \tag{6}$$

Referring to Table 5, the dice coefficient has improved compared to the training result. The accuracy has dropped by approximately 0.1 as the model might still need to be improved in order to identify vein pixels especially in fork vein branches besides the main vein branch. The model has achieved a relatively high specificity, indicating that it is highly capable to predict non-vein pixels correctly. It can be deduced that the model can effectively minimize misclassification of background pixels, such as patient’s forearm or surrounding backgrounds, as vein pixels. Besides, the sensitivity has improved from 0.4856 from the original U-Net architecture to 0.7806, which explains that the capability of the lightweight U-Net model in detecting positive vein pixels has been greatly enhanced. The high precision also indicates that more than half of the positively predicted pixels are actually true vein pixels. A highly accurate model must have both high recall (sensitivity) and precision [23]. In our experiment, we have successfully improved the model to achieve both high sensitivity and precision by using the lightweight U-Net model.

5. Conclusion

This research project has applied deep learning algorithm to detect subcutaneous vein for intravenous procedures. The optimum lightweight U-Net model has been employed and modified in order to segmentate the vein feature and the predicted outcomes show a promising result with accuracy of 0.88, specificity of 0.99, sensitivity of 0.78, and precision of 0.99. The experiment outcomes show a promising result on subcutaneous vein segmentation, and it can be concluded that this algorithm improve the capability and performance of current venipuncture machine and reduce the error of wrong vein detection. Thus, it can be concluded that the proposed algorithm can be applied in the venipuncture machine to locate the vein for IV procedures. Since there was a lack of forearm vein datasets available, not to mention the forearm vein datasets with groundtruth, because most of the vein datasets were made up of palm veins or finger veins. To overcome the issue of limited biomedical data, the proposed model was first been trained on a supervised vein dataset and then been evaluated using an unsupervised vein dataset. For each unsupervised forearm images, 20 checkpoints were selected from random directions to evaluate the extraction outcomes of true and false vein pixels by comparing the values of the checkpoints between the unsupervised forearm images and the predicted outputs. However, the checkpoints are chosen manually and carried out by human validation, a bigger pool of dataset with the corresponding groundtruth is needed to further validate the robustness of the proposed algorithm in extracting NIR vein accurately. Future work would attempt to improve the accuracy of the proposed model by acquiring a larger dataset for training and evaluation of the network. This project’s next phase aims to further improve the performance by optimizing the CNN methods and tested with a bigger pool of dataset.

Acknowledgement

This research was conducted in Universiti Tunku Abdul Rahman (UTAR) under Faculty of Information and Communication Technology (FICT), Kampar, Perak.

References

[1] Beecham, G. B., & Tackling, G. (2021). Peripheral line placement. In StatPearls, Treasure Island (FL): StatPearls Publishing.

[2] Ahuja, D., Gupta, N., & Gupta, A. (2020). Difficult peripheral intravenous access: need for some light. In Medical Journal of Dr. DY Patil Vidyapeeth, 13, pp. 422-423.

[3] Das, R., Piciucco, E., Maiorana, E., & Campisi, P. (2019). Convolutional neural network for finger-vein-based biometric identification. IEEE Transactions on Information Forensics and Security, 14(2), pp. 360-373.

[4] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention (MICCAI), 9351, pp. 234-241.

[5] Fang, Y., Wu, Q., & Kang, W. (2018). A novel finger vein verification system based on two-stream convolutional network learning, Neurocomputing, 290, pp. 100-107.

[6] Boucherit, I. Zmirli, M. O., Hentabli, H. & Rosdi, B. A.. (2020). Finger vein identification using deeply-fused convolutional neural network. Journal of King Saud University -- Computer and Information Sciences.

[7] Song, J. M., Kim, W., & Park, K. R. (2019). Finger-vein recognition based on deep DenseNet using composite image. IEEE Access, 7, pp. 66845-66863.

[8] Noh, K. J., Choi, J., Hong, J. S., & Park, K. R.. (2020). Finger-vein recognition based on densely connected convolutional network using score-level fusion with shape and texture images. IEEE Access, 8, pp. 96748-96766.

[9] Liu, W., Li, W., Sun, L., Zhang, L., & Chen, P. (2017). Finger vein recognition based on deep learning. In 2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 205-210.

[10] Meng, G., Fang, P., & Zhang, B. (2017). Finger vein recognition based on convolutional neural network. In MATEC Web of Conferences, 128.

[11] Hong, H. G., Lee, M. B., & Park, K. R. (2017). Convolutional neural network-based finger-vein recognition using NIR image sensors. Sensors (Basel), 17, p. 6.

[12] Al-Johania, N. A., & Elrefaei, L. A. (2019). Dorsal hand vein recognition by convolutional neural networks: feature learning and transfer learning approaches. In International Journal of Intelligent Engineering and Systems, 12, pp. 178-191.

[13] Harshan, G. S., Rayudu, S., Bharadwaj, K. A., & Saravanan, K. (2020). Finger vein recognition using VGG-16 CNN algorithm. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9(8), pp. 112-114.

[14] Mohaghegh, M., & Payne, A. (2021). Automated biometric identification using dorsal hand images and convolutional neural networks. Journal of Physics: Conference Series, 1880, p. 1.

[15] Tao, Z., Zhou, X., Xu, Z., Lin, S., Hu, Y., & Wei, T. (2021). Finger-vein recognition using bidirectional feature extraction and transfer learning. In Mathematical Problems in Engineering.

[16] Zhang, L., Cheng, Z., Shen, Y., & Wang, D. (2018). Palmprint and palmvein recognition based on DCNN and a new large-scale contactless palmvein dataset. Symmetry, 10, p. 4.

[17] Marattukalam, F., & Abdulla, W. H. (2020). Segmentation of palm vein images using U-Net. In 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 64-70.

[18] Wang, P., & Qin, H. (2020). Paln-vein verification based on U-Net. In IOP Conference Series: Materials Science and Engineering, 806, pp. 12-43.

[19] Zeng, J., Wang, F., Deng, J., Qin, C., Zhai, Y., Gan, J., & Piuri, V.. (2020). Finger vein verification algorithm based on fully convolutional neural network and conditional random field. IEEE Access, 8, pp. 65402-65419.

[20] Lefkovits, S. and Emerich, S. (2022). Boosting Unsupervised Dorsal Hand Vein Segmentation with U-Net Variants. Mathematics, vol. 10, pp. 2620.

[21] Jalilian, E., & Uhl, A. (2019). Enhanced segmentation-CNN based finger-vein recognition by joint training with automatically generated and manual labels. In 2019 IEEE 5th International Conference on Identity. and Behavior Analysis (ISBA: Security), pp. 1-8.

[22] Afaq, S. & Rao, S. (2020). Significance of epochs on training a neural network. In International Journal of Scientific & Technology Research, 9, pp. 288-485.

[23] Weaver, J., Moore, B., Reith, A., McKee, J., & Lunga, D. (2018). A comparison of machine learning techniques to extract human settlements from high resolution imagery. In IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium, pp. 6412-6415.