4AI000, Machine learning for multi-physics modelling and design: Predicting surface tension from drop images

Stijn van Esch and Bram van Boven

(Dated: July 21, 2023)

This report will explain the usage of two subsequently structured CNN to perform tensiometry.

I. INTRODUCTION:

A. Background

Surface tension plays a crucial role in many soft materials. Would that specific soft material have another surface tension, the material would have reacted differently. For example, the foam layer of a beer would not be so nice and stable. Or the question of whether water is drinkable or not, if it contains other fluids or particles the surface tension would also change.

Knowing the surface tension is key to predict the behavior of soft materials or check if the water is drinkable. Previously this was done by using tensiometry by inverting the Young-Laplace equation:

$$\sigma(\frac{1}{R_1} + \frac{1}{R_2}) = \Delta p - \Delta\rho g z \quad (1)$$

Given the shape of an experimental image, the equation can be iteratively optimized to find the surface tension (σ) and the pressure difference (Δp), as seen in figure 11. The two radii R_1 and R_2 correspond to the vertical and horizontal radii that describe the shape of the droplet. Lastly, the z term describes the difference in potential energy.

The problem with this method is that inverting the Young-Laplace equation is computationally expensive and error-prone. This raises the question, are there possible less error-prone and faster methods to determine the surface tension of a pendant drop?

B. Challenge

Design a supervised ML algorithm to perform tensiometry. The main characteristic of a supervised ML algorithm is that the algorithm needs labeled data. This labeled data can be gathered by a supplied script, that generates R_1 and R_2 coordinates given a certain surface tension σ.

The holy grail solution for this algorithm would be to get the surface tension from an image that is taken from a smartphone. To get to this solution the ML algorithm needs to be trained with photo-realistic data. This data can be gathered by coupling the supplied script, which generates R_1 and R_2 on a given σ, and a 3D rendering program. This way a large labeled dataset can be relatively easily realized. See figure 13 for the visualization.

II. PIPELINE:

As the challenge described in chapter I a supervised ML algorithm should be designed. The problem at hand can be described as a regression problem, having an input and wanting a continuous value as output. Many different supervised learning algorithms would be applicable to such a problem, linear regression, decision tree, random forest, support vector regression, and neural network. Each algorithm works perfectly fine, however, the problem at hand is complex because we have an image as input. Therefore a neural network would be the most sensible choice.

Because neural networks are mostly black boxes, the problem was split into two halves. This is to see if the preliminary results are as desired. As the Young-Laplace equation 1 suggests, most information arises from the shape of the drop. Image segmentation will be applied to get this information from a photo-realistic image. Subsequently, the detected drop is used as input for the second network, this network will predict the surface tension.

The first network that will be used is a U-net. A U-Net is a specific type of convolutional neural network (CNN) architecture that is commonly used for image segmentation tasks. One is, Natural Scene Segmentation: U-Net has been used for segmenting objects and regions of interest in natural scene images. It is also a widely popular structure in the medical scene, for example in recognizing patterns in CT images [1]. It has been applied to tasks like semantic segmentation, object instance segmentation, and boundary detection. This application is applicable to this challenge.

Training of this network will be done by using photo-realistic images as input and the true mask as the label.

The second neural network that is used is also a convolutional neural network that can capture spatial information and pixel relations. This is needed when the shape of the droplet needs to be interpreted. The architecture should aim to extract hierarchical features from the input image and learn discriminative representations for regression.

Training of this network will be done by using the true mask as input and the true σ.

Coupling the trained networks would result in the following pipeline:

Network	Input	Label	Output
U-net	Photorealistic image	True mask	Generated mask
CNN	Generated mask	True σ	Predicted σ

TABLE I. Pipeline for predicting .

III. U-NET

A. Design

The structure of the U-net is able to extract the most relevant features of an image that are important enough for recreating its mask. These features are extracted during the down-sampling phase, which can also be referred to as the encoder phase. It aims to remove all unnecessary info for computing the mask in the end.

After the down-sampling phase, which can also be referred to as the decoder phase, the important features are up-sampled again. This results in the generated masks.

While extracting all the relevant features from the images, the spatial information is lost. This is why there are so-called 'copy and crop layers' in the network to act as a guide on how the spatial information should be retrieved back during up-sampling.

Note that, in the beginning, the U-net does of course not know what the relevant features are. It is learning these parts based on the provided label, being the true mask.

During training the output of the U-net is compared to the true mask. Based on the Mean-Squared-Error between these two images a loss function is defined. While training it attempts to minimize the loss by adjusting the weights, i.e., learning the relevant features from the input image.

Now there is a better understanding of the working of the U-net. There is elaborated on the training of this U-net in the next few sections. The approach for training is to first take a dataset that is considered to be simple. If the result of this simple dataset is close to the true mask then there is moved on to a more complex dataset.

B. Simple Dataset

1. Data

First, to discover that the approach has potential, a simple dataset has been used for training the U-net. A sample of this set is shown in Figure 14.

This dataset contained 510 images in total, with only variations in the angle of the droplet. With each image, the true mask is also provided to the model to act as a label for what the output really should be.

2. Training

After training the U-Net on this dataset, the loss curves of Figure 1 are obtained. The loss is defined as the mean squared error between the true mask and the computed mask. From these losses, there can be noticed that the curves decrease a considerable amount. In the last 10 epochs, it seems to be still decreasing, however, since a log scale is used on the y-axis, this decrease is not that significant anymore, it is up to an order of 10^{-3}.

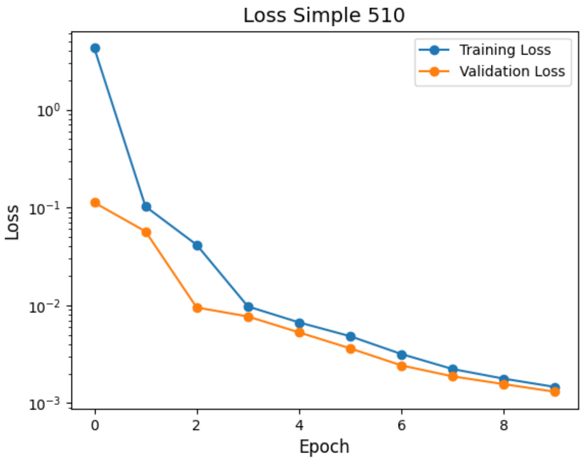


FIG. 1. Loss curves of the simple dataset

3. Results

Visualizing the output of the model based on a test image, a generated mask can be computed. Such a mash is shown in the following graph.

Since the above result is obtained for a dataset of 510 samples only. This way of computing the mask seems a potential way for doing this for more difficult datasets as well. This is explained in the next section.

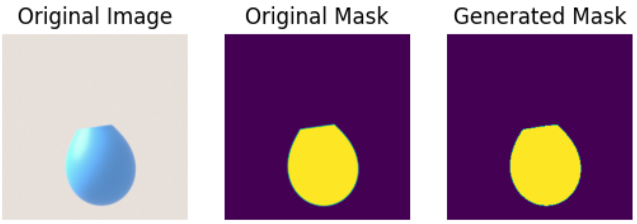


FIG. 2. The true image, true mask, and the generated mask respectively for a random image from its test set.

C. Advanced Model

1. *Data*

For the advanced model, more complex images are taken of the droplets. A realistic background is used, which has many variations. Next to that, the droplet material is improved significantly to really mimic the appearance of a water droplet. An example, of such an image, can be seen in Figure 15.

In the new representations of the droplet and its surrounding, there can clearly be seen that the image complexity has improved significantly.

For training, there are in total four different datasets created all varying in size. Again, the only differences are changes in the angles of the droplet. The different datasets have a size of 102, 510, 1040, and 2040 respectively.

2. *Training*

The U-Net is trained on each of the datasets. Each dataset is split for a training, validation, and testing set, with their percentage of the whole set being 60, 20, and 20 percent respectively. Each training session consisted of 10 epochs. In this section, the results of the losses are explained in more detail.

The behavior of the losses is shown in Figure 16, Figure 17, Figure 18 and Figure 19, for each dataset respectively.

Clearly, a distinction can be made between the two smallest and two largest datasets. For the smaller two datasets, no decrease is present and the loss rather stays the same. However, the contrary is true for the latter two datasets. In these loss curves, there is a clear decrease noticeable over the range of epochs. The decrease in the loss for the last two datasets might be due to the fact that the network is more able to adjust its weights since it is seeing more data. Later in this report, the outputs of all four networks are shown and there is reflected back on this distinction between the losses.

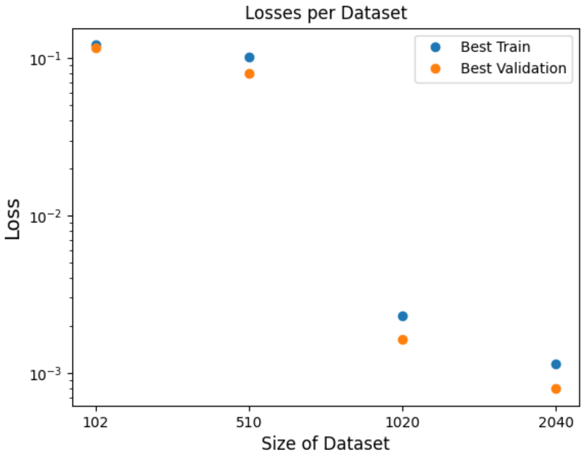


FIG. 3. Losses of each training session.

In Figure 3, the best losses of the training sessions for each size of the dataset are shown. Noticeably, the losses of the first two datasets are high compared to the last two larger datasets.

The training on the dataset consisting of 1020 samples already shows a better decrease in loss. Lastly, the dataset with 2040 samples in total has an even larger decrease in its loss and therefore is expected to perform best in creating the corresponding mask.

The results will be shown and explained in the next section. Also, a measurement is used on how accurately the mask is computed.

3. *Results*

The models trained from the previous section are tested on the test set that is split from the total set. Since, in some cases, the test set does not contain the whole range of the surface tensions where is trained on, it is hard to predict how it would perform on the unseen surface tensions in the test set. However, it is still best to use the test set, since the model has never seen this data. In the testing of the models, there is computed how accurately the masks are recreated. In Figure 4, a visualization of the generated masks is given for each model.

As can be seen in the shown outputs of each model, the first two models have low performance in predicting the mask. The third model already has improved quite a bit, meaning it is able to partially predict the droplet well. The last model, trained on the largest dataset, is able to predict the droplet even better as its shape is

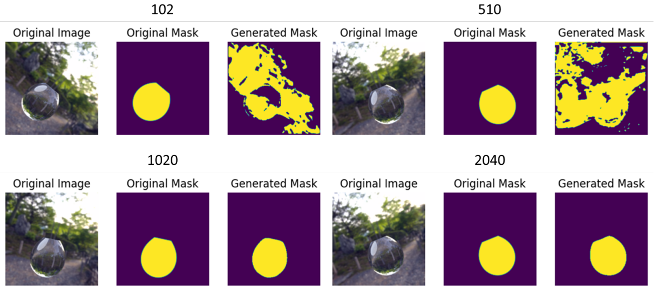


FIG. 4. Results of the generated masks per model for a random image from the test set.

more like the droplets and only the outer edge is slightly more unclear compared to the real mask.

However, the generated mask visually seems to improve, a better approach is to compute the performance of the mask generation based on Pixel Accuracy (PA). The PA is performed by checking whether a pixel from the true mask is generated correctly by the predicted mask. Subsequently, the amount of correct pixels is divided by the total amount of pixels in the image.

Since, it is most interesting to look at the contour of the droplet, meaning the background is not relevant for computing the surface tension eventually. The PA is only focused on the pixels where the droplet is located in the true mask, subsequently, all these pixels are checked in the generated mask and there is computed how many of them are classified correctly. By using this way of computing the PA, there needs to be checked visually whether the mask is somewhat of a droplet. This occurrence will be seen in the next paragraph.

The PA is performed on the whole test set for each model and the results are shown in Figure 5. For the smaller datasets, the test set is also smaller than for the larger models, therefore the data points don't have an equal amount in Figure 5. Critically looking at the results of the models consisting of 102 and 510 samples. There can be seen that they are underperforming a lot, which is also already visible in Figure 4. Although model 510 has a reasonably high accuracy, this is likely to be more luck than a correct prediction. As there is now only checked on the shape of the droplet, and not on the rest of the background.

However, looking at the largest two networks there can be seen that the accuracy is improved significantly, which is also expected to happen based on the visualizations of these masks.

The results of model 2040 are also as expected the best. The accuracy is even higher than the model 1020. Especially, in Figure 6 there can be noticed that the range for the surface tension is from 30 up to 80 in $\frac{mN}{m}$. Although the physical range for a water droplet is between 40 and 70, the range is artificially extended. This is done for the reason that at the edges of the range, the neural networks tend to perform less, as it is in the end 'interpolating' between the values it has learned during training. This effect is visible in Figure 6 as the performance towards the edges is decreasing. Therefore, by artificially extending the range, the model performs better in the physical range where it will be used.

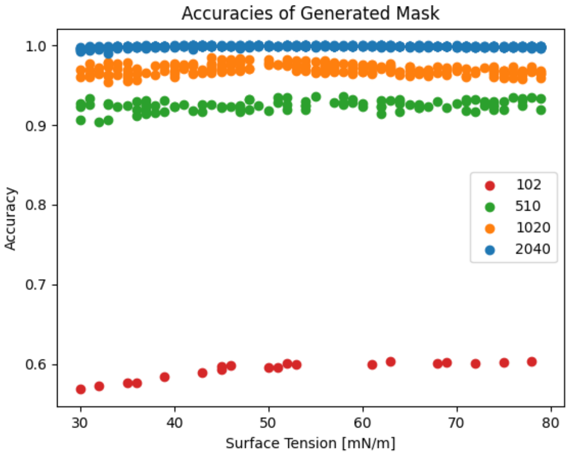


FIG. 5. Results of the Pixel Accuracy of all models.

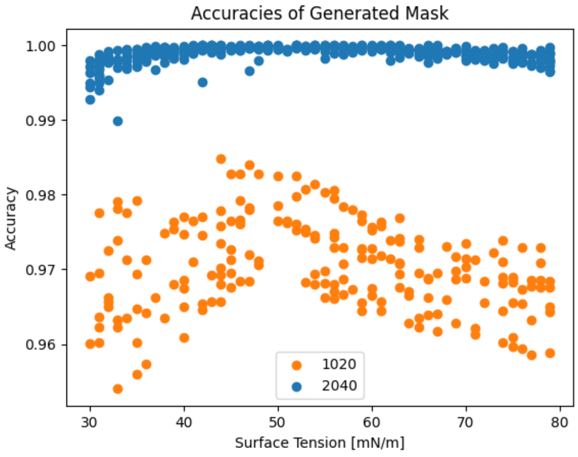


FIG. 6. Results of the Pixel Accuracy of the largest two models.

IV. CNN

A. Design

The design of the neural network should be able to process an image of $w \times h \times c$ (tensor), where channel = 1, because the output of the U-net is a 1-channel binarized tensor. The output, given the input tensor, should be a 1×1 scalar value as the value should be the predicted surface tensions, which is a continuous value. The design should carefully be tested, as a too complex network can introduce overfitting but making the network too simple can negatively impact the performance of the network.

The structure of the network will be:

- Convolutional layer
- Pooling layer
- Fully connected layer

The final design is visualized in Figure 20. The network consists of a convolutional layer followed by a pooling layer, which is followed by two more of these pairs and at the end there are three fully connected layers. The design is able to handle a 512×512 sized tensors and generate a scalar, which is the predicted value of the surface tension. The activation function that is used is a leaky ReLU, to counter the vanishing gradient problem.

B. Data

Data used for training the CNN are the true masks (as seen in Figures 2 and 4 combined with the corresponding surface tension. Multiple datasets are created to both train the model and visualize the relation between the size of the dataset and performance. The dataset sizes that are used are: 510, 1020, 2040, 4080, 8160, 10200.

All sets are split, to prevent bias visualizations and spot overfitting. As the performance should be measured on a set that was not seen before by the network. The sets are split in 60% training, 20% validation and 20% testing.

C. Training

All 6 datasets are used to train the CNN individually and subsequently evaluated. The loss graphs for each set can be found in Figures 21 up to and including Figure 26. To compare the effects of the size of the training sets the best validation and training losses are plotted, note that best means lowest. The loss is defined as the average MSE for a particular epoch.

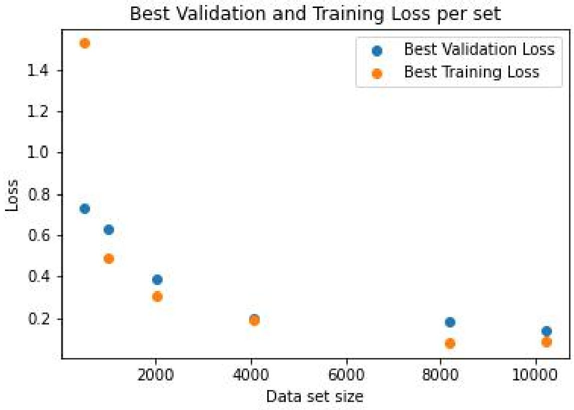


FIG. 7. Best losses per dataset size

In Figure 7 a relation can be seen between the size of the dataset and the losses, increasing the size lowers the loss. This is as expected. Another important part to notice is, that the validation loss is slightly higher in general which indicates that the network is not overfitting.

D. Results

The losses seen in Figure 7 are relative and can not conclude a lot about the performance of the network. Therefore the accuracy of the network should be evaluated. A hit is defined by:

$$\epsilon = |\text{real value} - \text{predicted value}| <= 0.5, \qquad (2)$$

resulting in the following graph:

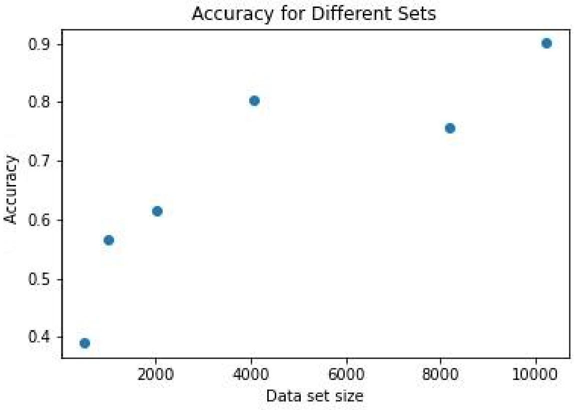


FIG. 8. Accuracy per dataset size

The relation that can be seen in Figure 8 is as expected, increasing the dataset size increases the accuracy. Note that the values on the horizontal-axis are the sizes of the total set. For accuracy testing the test set is used. So, a dataset size of 10200 would contain 2040 testing images.

For a total dataset size of 10200, the trained CNN has an accuracy of 90%, combined with the clear positive relation between dataset size and accuracy increasing the dataset size would most likely result in an even higher accuracy. Therefore the CNN can be concluded to be successfully designed.

V. COMBINING THE NETWORKS

In Table I the structure of the pipeline is discussed, this pipeline is used to predict the surface tension from a given image. The evaluation is done by creating a new unseen test (size = 408) and analyzing the images with the network. The output, 408 predicted surface tensions is compared to the 480 labels. This is the test set generated when training the U-net.

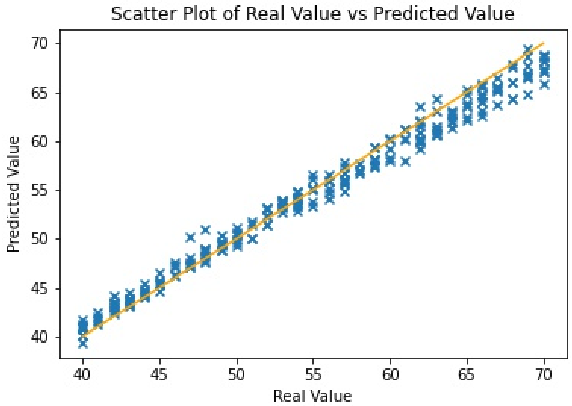


FIG. 9. Scattered predictions of the surface tensions in $\frac{mN}{m}$

In Figure 9 a plot is seen which visualizes all the predicted surface tensions with the corresponding label. The diagonal line that is seen across the plot would be the perfect prediction. What can be stated from analyzing Figure 9 is that there are no outliers. However, the predictions become less accurate when the real surface tensions are increased. This effect can be explained by the physical boundary that arises from the fact when the surface tension is at the higher end of the physical limitations the droplet is round. When increasing the surface tension even further, the change in the size of the drop is minimal. Therefore the network can harder differentiate between the droplets. This effect can be seen more clearly in Figure 10

The average MSE that results from all predictions is MSE = 4.06, mainly caused by the predictions at the higher end. For the total accuracy, the same definition is used as in Formula 2. Resulting in an accuracy 21.32%. Depending on the application, the hit definition can be changed, resulting in changing accuracies. These accuracies can be seen in the following table:

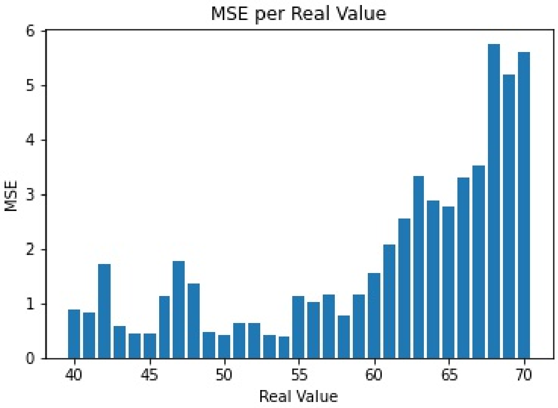


FIG. 10. MSE per real value

Hit ($\epsilon <=$)	**Accuracy** %
0.5	21.32
1.0	41.91
1.5	60.78
2.0	74.02

TABLE II. Accuracy per hit definition

VI. CONCLUSION

To conclude, the U-net is shown to work. Important to note is that in order to reach the desired accuracy of the U-net that enough data is provided. As is shown in this report, providing more data results in a more accurate mask.

The same is true for the second network providing the trend of the accuracy improving when more data is given to the model while training.

By combining the two models, so using the generated output from the U-net and feeding it into the CNN the best accuracy obtained is 21.32% percent, for a hit definition of $\epsilon <= 0.5$. The accuracy is also dependent on the maximum deviation of the generated surface tension and the true surface tension. One can imagine that for different applications a more or less strict deviation is allowed, which also tells when or in which application the model would behave as good enough.

VII. DISCUSSION

A general note to improve both models is to increase the size of the dataset. In this report, there is shown that for both models the accuracy increased if the size of the dataset is increased as well, see Figure 5 and Figure 8. By improving both models individually the accuracy of the combined model is then also expected to improve.

As shown in Figure 5, the Pixel Accuracy for the best trained U-net is rather high. However, there should be noted that the mask is the weakest point in this structure of the sequential neural network. Although the PA is for the last trained model considerably high there should be noted that the accuracy that is lost, is due to the wrongly classified pixels at the surroundings of the droplet. Since the CNN is using the edges of the droplet, this is the only relevant information that is changing in different droplets. Most of the accuracy that is lost if the models are combined, is due to the mask not being computed correctly by the U-net.

Currently, the size of the generated mask is 256×256. If this size is increased to for example 512×512 or even1024×1024 the accuracy of the generated mask is likely to improve, since the edges will be computed more accurately.

Another point of improvement that leads to a better generalization is to also train the model on different kinds of volumes and sizes of needles. In this report, the only parameter changed is the surface tension.

Lastly, the final model shown in this report is solely trained on the images in Figure 15. Since it has only seen this type of background it is expected to not generalize well in other environments. Therefore, if it is desired to use this model in more different cases it should be trained on a richer dataset, i.e., one containing different backgrounds. In general, it is advised to think about a desired environment and train the model on that different situation to avoid the need of having a dataset containing all possible backgrounds, since this is an unrealistic way of gathering data.

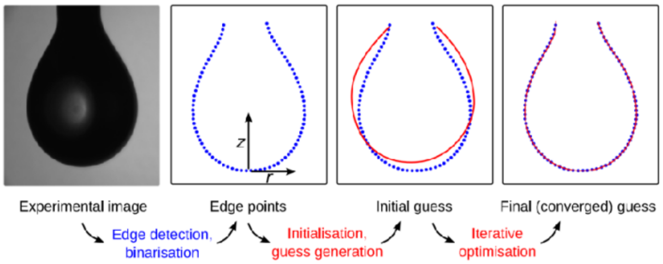
[1] G. Du and X. Cao and J. Liang and X. Chen and Y. Zhan, Medical image segmentation based on U-Net: A review, Journal of Imaging Science and Technology **64(2)** (2020).

[2] J.D. Berry and M.J. Neeson and R.R. Dagastine and D.Y. Chan and R.F. Tabor, Measurement of surface and interfacial tension using pendant drop tensiometry (2015).

[3] Z. Yicheng and N. Jaensson, Bayesian PendantDrop Tensiometry (2023).

[4] O. Ronneberger, P. Fischer, and T. Brox, U-net: Convolutional networks for biomedical image segmentation, in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, LNCS, Vol. 9351 (Springer, 2015) pp. 234–241, (available on arXiv:1505.04597 [cs.CV]).

Appendix A:



J.D. Berry et al. *J. Colloid Interface Sci*, 2015.

FIG. 11. Tensiometry steps [2]

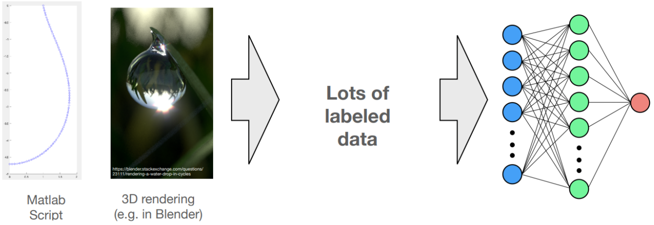


FIG. 12. Visualization training data

Appendix B:

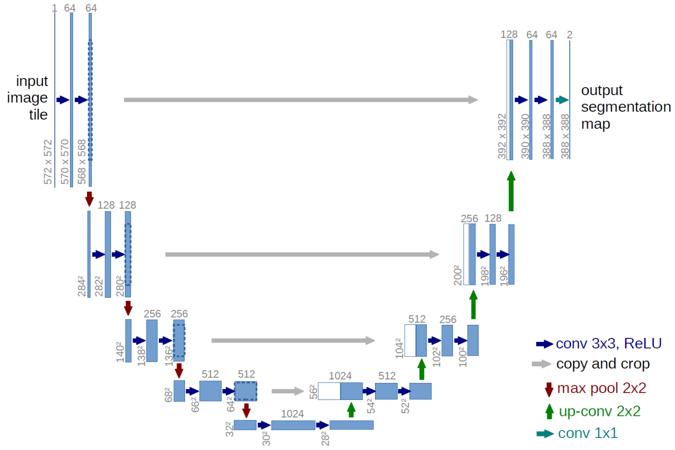


FIG. 13. Visualization of the U-net [1].

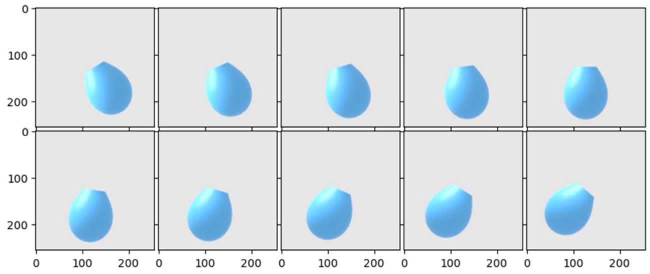


FIG. 14. Samples of the simple dataset with on the axis pixels (256 × 256).

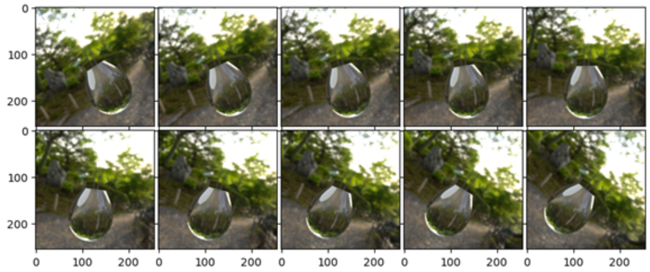


FIG. 15. Visualization of the realistic droplet with on the axis pixels (256 × 256).

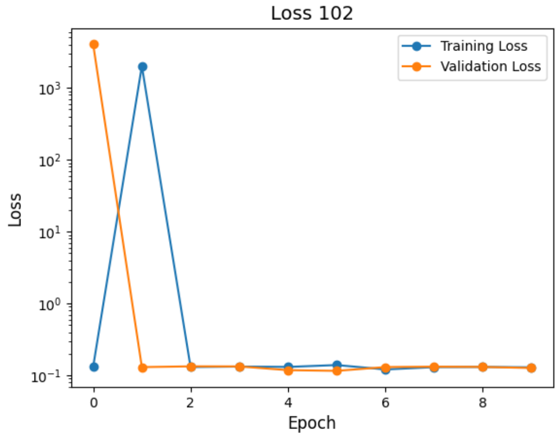


FIG. 16. Loss curves of training with the dataset of 102 images.

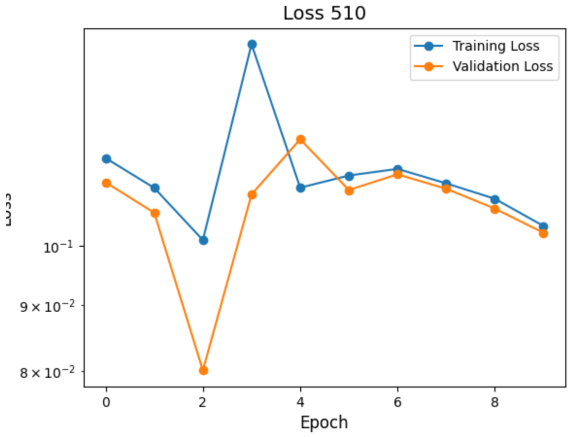


FIG. 17. Loss curves of training with the dataset of 510 images.

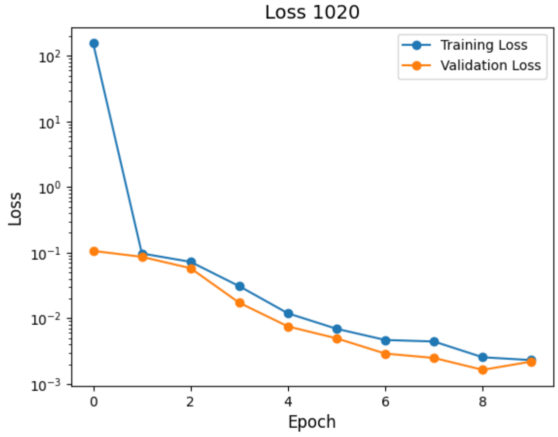


FIG. 18. Loss curves of training with the dataset of 1020 images.

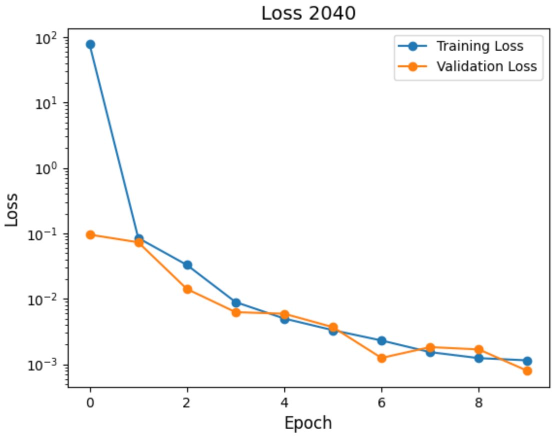


FIG. 19. Loss curves of training with the dataset of 2040 images.

Appendix C:

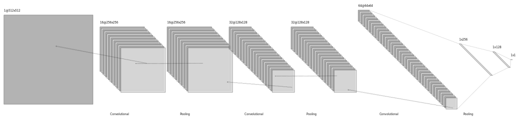


FIG. 20. Design of the CNN

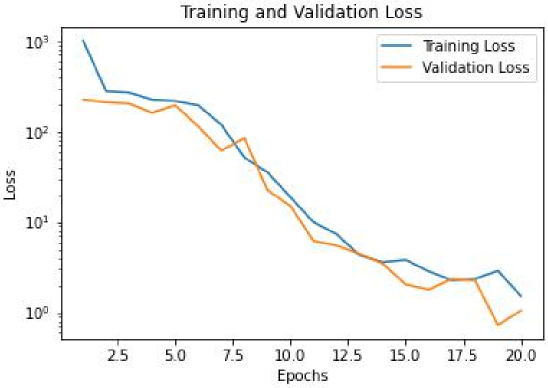


FIG. 21. Loss plot 510

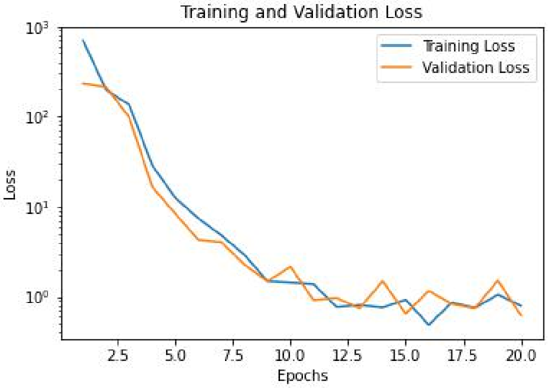


FIG. 22. Loss plot 1020

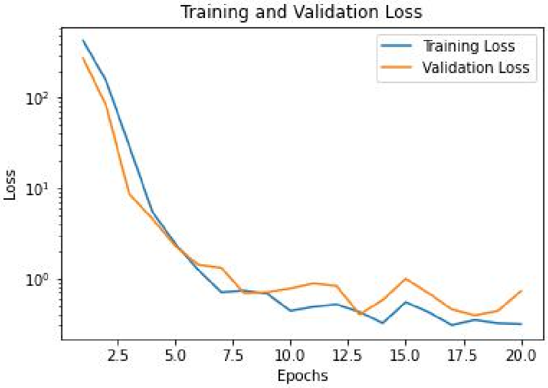


FIG. 23. Loss plot 2040

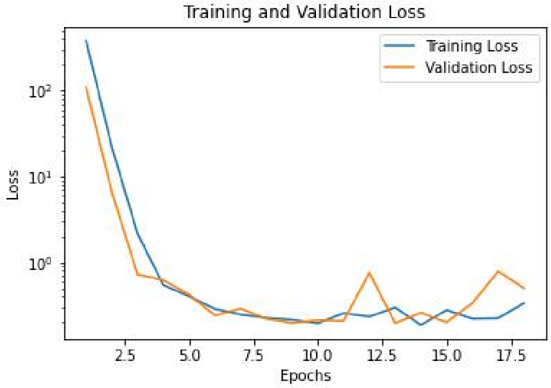


FIG. 24. Loss plot 4080

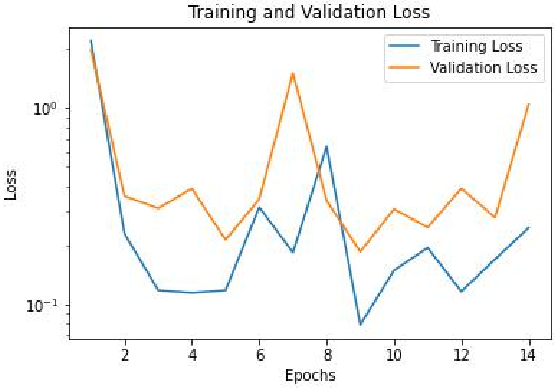


FIG. 25. Loss plot 8160

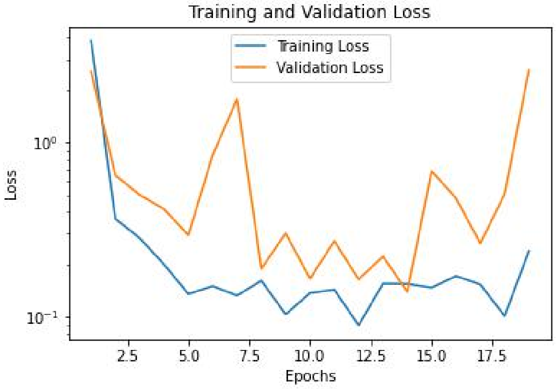


FIG. 26. Loss plot 10200