Wound Stage Classification using Few-Shot Learning

A Theoretical Framework for Spatio-Temporal Attention Few-Shot Learning

1st Narges Norouzi
Computer Science and Engineering
University of California, Santa Cruz
Santa Cruz, CA, USA
nanorouz@ucsc.edu

2nd Alex Salman
Computer Science and Engineering
University of California, Santa Cruz
Santa Cruz, CA, USA
aalsalma@ucsc.edu

I. Introduction

Clinical treatment could be advanced with accurate classification to wound stage. Clinicians could assign the right treatment at the right time to accurately classified wound stage. This example helps motivate our work.

Processing wound images has been revisited very recently with the aim to use deep learning algorithms to detect wound and to give it's size estimation [1]. Furthermore, Few-shot learning has recently received a lot of attention in video classification due to its potential to reduce video annotation cost significantly [2]. Few-shot classification aims to learn a classifier to recognize unseen classes during training with limited labeled examples [3]. In fact, the goal of few-shot learning is not to let the model recognize the images in the training set and then generalize to the test set. Instead, the goal is to learn and to know the similarity and differences between objects [4].

In this work, we are classifying frame(s) of wound into one of four classes (stages of wound healing) using Meta-Learning Framework. The framework takes input from a general video spatial and temporal representation using 2D-CNN + LSTM and 3D-CNN. Then, to make the most out of these representations, we use self and cross attention mechanisms to highlight the critical spatio-temporal areas in the wound frames. At the end we apply a lightweight fusion network and a nearest neighbor classifier to classify each query of the frames.

We are following the implementation of a previous work on classification using spatio-temporal attention few-shot learning [1] and using a dataset of C57BL/6J mice [2]. In this work we proposing a theoretical framework as a new approach to estimate wound stage for future research and application usages.

II. Data

We are using the images of mice wound for training and testing our models. These images are time series of photos during 15-day wound closure process of C57BL/6J mice. It consists of a total of 255 photos. The C57BL/6J mice photos are of 4 young (12-14 weeks old) and 4 aged (22-24 months old). The daily wound photos were captured with a cell phone at a fixed distance of 12 cm from day 0 (the surgery day) to day 15 (the experimental endpoint) [1]. Each mouse wound is captured twice a day, left and right photos. One of the images is missing, so the total is 255 images. These images were annotated by 10 students and aggregated in the four classes (labels) representing the wound stages (*hemostasis, inflammatory, proliferative, and maturation*).

III. Data Processing

As we are working on spatio-temporal attention few-shot learning mechanism, we need to process multiple mouse frames (images) to learn general then better object representations from the frames. Subsections A, B, and C are the data pipeline from raw images to model input:

A. Augmentation

We have generated a set of videos in different lengths of wound frames consecutively. For each mouse of the eight mice, we created 270 videos from length of one frame to sixteen frames for both of the left and right side captures. For a length of three and Right side images as an example, the appended frames could be Day 2, Day 3, and Day 4, representing the same mouse same side frames captured on the three consecutive days. However, we padded these videos with black frames from left and/or right ends of the the mouse frames to ensure wound stage consistency in training and testing. For the three days example, we padded two black frames at the left side of the video as for Day 0 and Day 1 and eleven frames at the right side for Day 5 through Day 15. The total is 2160 videos of length 16 frames each. We generated these *.mp4* videos with the same name of the last mouse image appended to the mouse frames. We use the file name in mouse number and mouse side selections in the next two steps below, normalization and data split.

B. Normalization

We are using OpenCV to read all the 2160 videos of the four stage classes. For the images, we add the video

frames in an array, resize them to frame size of 128*128 pixel and depth of 16 frames to have a shape of 128*128*16, Width(W)*Height(H)*Depth(D). At the end, we return them as a NumPy array with pixel values normalized between 0 to 1. And to label these categorical data, we used Keras utility, to_categorical to convert a class index, integers (0, 1, 2, 3) as the four classes of our wound stages to a binary class matrix. That has been done separately for the labels of training, validation and testing.

C. Data Split

We have split our data on 75% for training and 25% for testing and validation:

- *Training data* is of six mice total, 1620 videos, three aged and three young.
- *Validation data* is of the right side captures of an aged mice and a young mice, total of 270 videos.
- *Testing data* is of the left side captures of the same validation data mice, total of 270 videos.

See Table I below for breakdown and mice numbers:

TABLE I
C57BL/6J MICE DATA SPLIT

Aged and Young (R+L)*	**Split Distribution**		
	Training	***Validation***	***Testing***
A8-1	R + L		
A8-3	R + L		
A8-4		R	L
A8-5	R + L		
Y8-1	R + L		
Y8-2	R + L		
Y8-3	R + L		
Y8-4		R	L
Total videos	1620	270	270

*Note that each side of a mouse, Right and Left has 130 videos.

IV. MODELS AND ALGORITHMS

Now that we have the raw images processed, the data is ready for our first module, **Embedding module**. As we are following MASTAF model [2], the goal of the embedding module is to learn the Spatio-temporal representation for each video. So we used two video classification models as the spatio-temporal embedding module:

A. *2D-CNN + LSTM*

This combination of 2-dimensional CNN and LSTM identify features of the data while keeping into account the temporal relation. For video classification, this approach effectively captures the Spatial relation on the individual frames and the temporal relation across the different frames. As a result of this convolution structure, the ConvLSTM is capable of taking in 3-dimensional input (W, H, num of channels), whereas a simple LSTM only takes in 1-dimensional input hence an LSTM is incompatible for modeling Spatio-temporal data on its own [5].

With input shape of (16, 128, 128, 3) as D, W, H, and three is the number of channels (RGB), we built a sequential model using Keras. The model consists of five ConvLSTM2D layers and one LSTM layer [6] [7]. For full model, check out code on the project repository on Github[1].

We trained the model on batch size of 10 and for 10 epochs to avoid over-fitting. Then we ***fine-tuned*** the model on a very low learning rate, *1e-5*, using Adam optimizer.

B. *3D-CNN*

In 3D-CNN (aka. Slow Fusion), the temporal and spatial information are merged slowly throughout the whole network that is why it is called Slow Fusion.

With the same input shape of 2D-CNN + LSTM , (16, 128, 128, 3), we built a sequential model using Keras. The model consists of four Conv3D layers. Please, check out our code with the model summary on the project repository on Github[1].

We trained the model on batch size of 4 and for 7 epochs to avoid over-fitting. Then we ***fine-tuned*** the model on a very low learning rate, *1e-5*, using Adam optimizer.

V. RESULTS

We evaluated the model on the test data (features and labels). On *2D-CNN + LSTM*, we got test accuracy of 69.12% and loss of 1.3252. When fine-tuned, we got about 5% accuracy increase, 74.63%, and about the same loss, 1.3042. On *3D-CNN*, we got higher test accuracy, 79.78%, and lower loss of 0.6865! When fine-tuned, we got about 2% accuracy increase, 81.99%, but higher loss, 0.7368. Test accuracy of models are in Table II below and total accuracy and loss are in figures 1, 2, 3, and 4 below.

TABLE II
MODELS ACCURACY

Architecture	**Test Accuracy**	
	Baseline	*Fine Tune*
2D-CNN+LSTM	69.12%	**74.63%**
3D-CNN	79.78%	**81.99%**

While designing both of the models' architecture, we observed that it is very critical to have only 16 frames videos. Small models we have tried with about 100k parameters can only result in test accuracy of about 65%. Therefore, we had to increase the size of models to achieve the current accuracy.

VI. NEXT STEPS

As we created the general representations of features, the next step will be using a *Self-attention module* to highlight the critical information in the representation of each class, and *Cross-attention module* to highlight the critical Spatio-temporal region in the representation itself. The step after is to use the *Attention fusion module*: First we compute the probability of predicting the class of features using both *Self-attention representation* and *Cross-attention representation*. Second, to take advantage of the discriminative information from two attention mechanisms, we leverage the attention fusion module with the nearest neighbor classifier. The step

[1]https://github.com/alexsalman/CSE247

after will be conducting *Multi-task training* to reduce the risk of over-fitting by regularizing the embedding network [2].

REFERENCES

[1] Yang, Hsin-ya; Bagood, Michelle; Carrion, Hector; Isseroff, Rivkah (2022), Photographs of 15-day wound closure progress in C57BL/6J mice, Dryad, Dataset, https://doi.org/10.25338/B84W8Q.

[2] Liu, Rex; Zhang, Huanle; Pirsiavash, Hamed; Liu, Xin (2022), MASTAF: A Model-Agnostic Spatio-Temporal Attention Fusion Network for Few-shot Video Classification. University of California, Davis. https://arxiv.org/pdf/2112.04585.pdf.

[3] Chen, Wei-Yu, et al. "A Closer Look at Few-Shot Classification." Google Sites, 2019, https://sites.google.com/view/a-closer-look-at-few-shot/.

[4] Thailappan, D. (2021, May 1). Few-Shot Learning — An Introduction to Few-Shot Learning. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/05/an-introduction-to-few-shot-learning/

[5] Anwar, Taha, and Rizwan Naeem. "Human Activity Recognition Using Tensorflow (CNN + LSTM)." Bleed AI, 17 Nov. 2021, https://bleedai.com/human-activity-recognition-using-tensorflow-cnn-lstm/.

[6] Donahue, Jeff, et al. "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description." ArXiv.org, 17 Feb. 2015, https://arxiv.org/abs/1411.4389v3.

[7] Woodfrog. "Woodfrog/ActionRecognition: Explore Action Recognition." GitHub, 11 Mar. 2017, https://github.com/woodfrog/ActionRecognition.

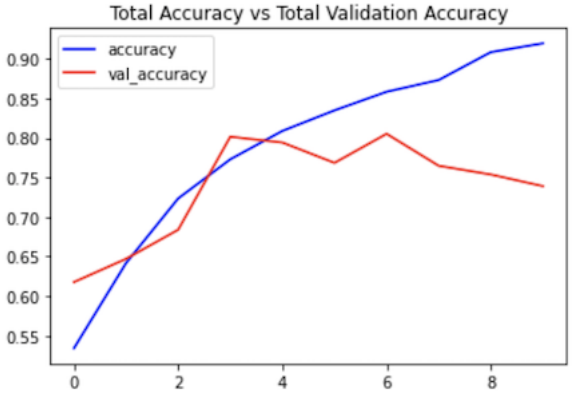

Fig. 1. 2D-CNN+LSTM Accuracy.

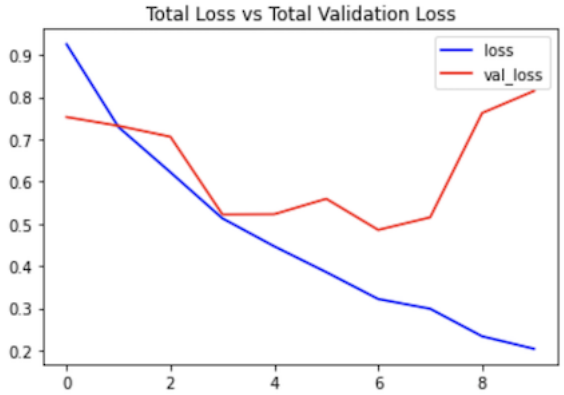

Fig. 2. 2D-CNN+LSTM Loss.

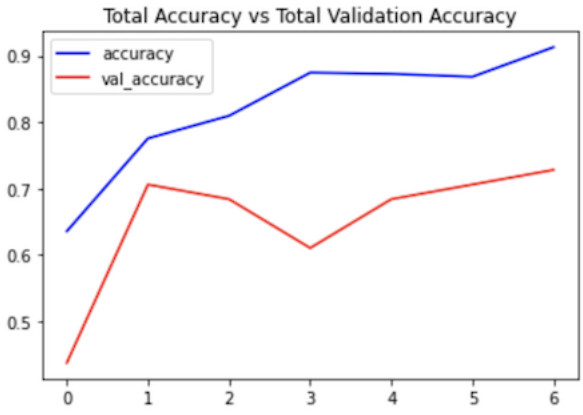

Fig. 3. 3D-CNN Accuracy.

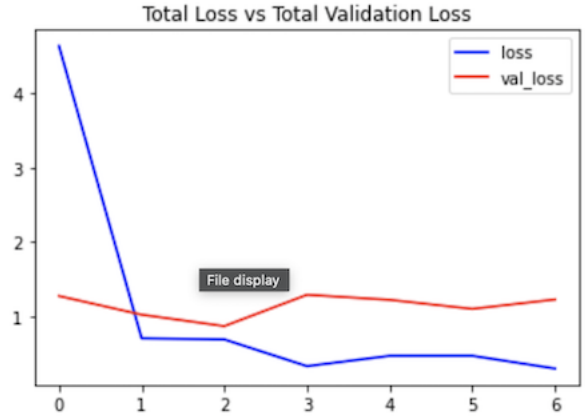

Fig. 4. 3D-CNN Loss.