Comparing Two Facial Pain Detection Algorithms

Zhimao Lin, Christopher Demario Sastropranoto, and Yasamin Zarghami
University of Toronto
{zhimao.lin, christopher.sastropranoto, yasamin.zarghami}@mail.utoronto.ca

Abstract

Pain detection in cognitively impaired patients can be detected through the use of neural networks. Razaei et. al (Alias name: Main paper) and Xu et. al (Alias name: ExtendedMTL4Pain paper) both provide different architectures to solve this. It is found that when both models are applied to a new dataset, that the Main paper is slightly more effective in predicting PSPI scores. It is found that the test MSE for the Main paper is 0.237, Pearson correlation is 0.972, F1 score is 0.838 with a training time off 24.04hrs compared to an test MSE of 0.145, Pearson correlation of 0.954, and F1 score of 0.835 with a training time of 7.41hrs for the ExtendedMTL4Pain paper. We conclude that the Main paper is more suitable for medical professionals because of the slightly higher Pearson correlation and F1 score, as well as the method in which the test data is constructed.

1 Introduction

Pain level diagnosis is a huge problem for medical professionals when caring for patients who have difficulty communicating [7]. Numerous proposals have been made, such as [6] uses of Relevance Vector Regression (RVR) with discrete cosine transform (DCT) to predict the pain score. Surprisingly, Long short-term memory (LSTM) [12] can take advantage of the temporal relationship between video frames to detect pain. Also, Openface library uses Support Vector Regression (SVR) to predict the pain score [2].The Main paper, [11] by Razaei et al., proposes a convolutional neural network (CNN) which combines reference and target images in its first layer and then feeds them to a network inspired by LeNet [4] to predict the PSPI score. The ExtendedMTL4Pain paper uses one iteration of VGGFace [10] and a VGG16 FC7 layer to measure the pain levels from images and accumulate them over videos before processing the videos inside a hidden layer to predict the PSPI score [15].

We aim to compare the effectiveness of the Main paper [11] and the ExtendedMTL4Pain paper [15] when trained on a new dataset from Kaggle [1], which contains 48373 images and corresponding PSPI scores. To have a fair comparison, the same pre-processing method is applied to the inputs for both models. Then, hyper-parameter tuning is performed based on randomly sampled 3000 training data and 150 test data. Then, the optimal hyper-parameters are used to train these models on the entire dataset. Finally, the mean squared error (MSE), Pearson correlation, and F1 score are calculated to evaluate those two models during the testing phase.

2 Architectures of the Two Papers

The Main paper develops a deep neural network that can detect pain from the faces of elders with dementia [11] using the UNBC-McMaster dataset [9]. By leveraging pairwise comparative inference and contrastive training, this model outperforms other state-of-the-art computer vision related pain monitors [11]. Pre-processed images are fed into the architecture as shown in Figure 1 to create

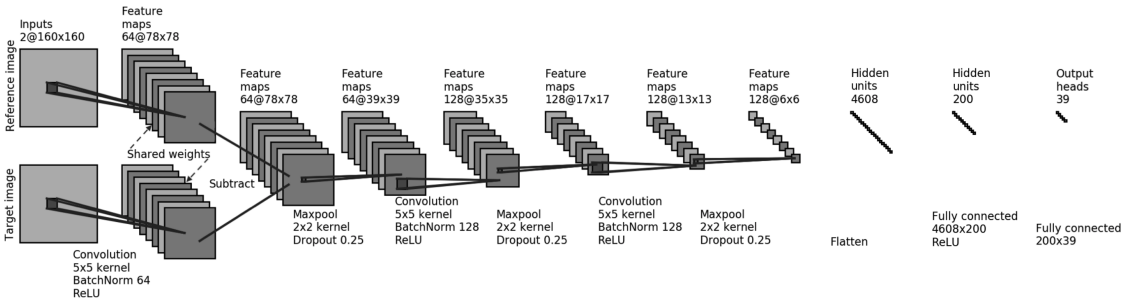

Figure 1: Figure showing the architecture of the Main paper Model [11].

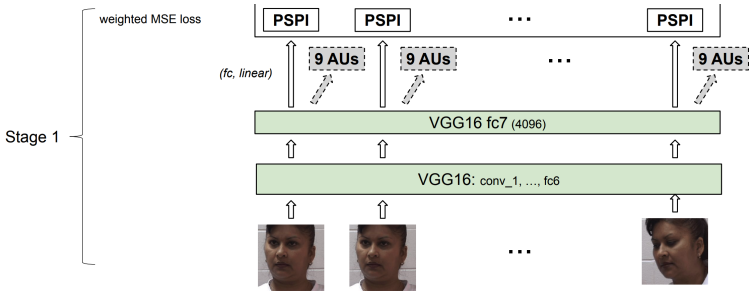

Figure 2: Figure showing the architecture of the ExtendedMTL4Pain model [15].

a target image (PSPI>0) and a reference image (PSPI=0) through a convolutional layer to create two feature maps. The maps are then fed into an architecture which resembles LeNet [4]. The ExtendedMTL4Pain [15] uses CNN to classify pain and non-pain from facial expression. Unlike the first model, this architecture uses the DPM Face Detector to pre-process the images which have high computational complexity [5]. Then, it predicts PSPI score, action units (AU) and visual analogue scales (VAS) scores in different stages. Due to the data limitation, only the first stage is reproduced. As shown in Figure 2, input images are fed into the VGG-16 architecture [10] and then the VGG16 FC7 layer.

3 Method

3.1 Pre-Processing Images

Firstly, the kaggle dataset [1] is written into a csv file, where each row contains the image path and its PSPI score. For the Main paper, another csv file is created based on the previous one where each row contains a pair of reference image (PSPI=0) and target image (PSPI>0) paths and the PSPI score of the target image. Each target image is paired with randomly sampled 5 reference images of the same person. Finally, to compare between those two models, patient's faces are detected, cropped and resized to 160 by 160 pixels using the S3FD model [13] and facial landmarks are extracted using the FAN model [3] for both models. During the pre-processing stage, there are 25 invalid images that fail the face detection, which are excluded for training and testing.

3.2 Hyper-parameter Tuning and Evaluation

3000 training images and 150 test images are randomly sampled from the 48373 images (due to hardware limitations). To evaluate the regression precision of the PSPI score, the average MSE loss and the Pearson correlation score are calculated during the evaluation on the test set. Also, to evaluate both models for a real-world binary classification problem (pain or not pain), F1 score is calculated, on the test set, with a threshold equaling to 2 (pain=PSPI>2) according to standard practice [8]. The hyper-parameters that are tuned for the Main paper model are the learning rates, dropout rate, batch

size, and the output size of second last fully connected layer (FC2 size). The hyper-parameters that are tuned for the ExtendedMTL4Pain model are batch size, learning rates, image scaling size before cropping, and weight decay.

Once the optimal hyper-parameters are found, both models are trained using the optimal hyper-parameters. The entire dataset is split into training set (80%) and testing set (20 %). Each model is trained on the entire training set for 5 epochs. Then, the models are evaluated by training time and calculating the average MSE loss, Pearson correlation score, and the F1 score (threshold is the same as hyper-parameter tuning) on the test set.

4 Experiments and Results

4.1 Optimal Hyperparameters and Training Loss

The optimal hyper-parameters for the Main paper model are 0 dropout, 100 for FC2 size, 0.0001 learning rate and 50 batch size. It is found that changing the dropout rate from 0.25 to 0 reduces the average MSE by around 65.7%, and the 0.25 dropout shows no correlation and 0 F1 score compared to around 0.371 and 0.622 for the Pearson correlation and F1 score respectively (see Appendix A for more results). The optimal hyper-parameters for the ExtendedMTL4Pain model are: resizing 200 pixels for before image cropping, 0.0005 weight decay, 0.001 last layer learning rate, 0.0001 overall learning rate and 50 batch size. It is found that increasing the batch size to 100 results in an out-of-memory failure due to the hardware limitation (please see Appendix A for more results).

Using the optimal hyper-parameters found in section 4.1, it is shown in Figure 3 and Figure 4 that the average training MSE per epoch reached about 0.103 and 0.154 for the Main paper and the ExtendedMTL4Pain paper respectively after 5-epoch training. Note that the fifth epoch represented as the fourth due to python's numbering system. For more details of average MSE loss of each batch in each epoch, please refer to the Appendix B.

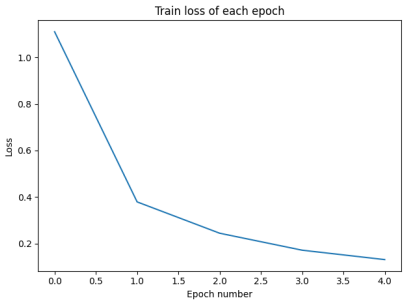

Figure 3: Main paper: The average MSE loss of each epoch.

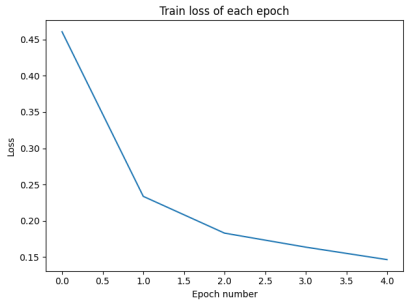

Figure 4: ExtendedMTL4Pain paper: The average MSE loss of each epoch.

4.2 Testing Metrics

After training the Main paper model and the ExtendedMTL4Pain model for 5 epoch on the training set, the evaluation results of those two models on the test set are the following:

Table 1: Table showing the evaluation metrics of the Main paper and the ExtendedMTL4Pain paper.

Metric	Main paper	ExtendedMTL4Pain paper
Average Mean Squared Error	0.237	**0.145**
Pearson correlation	**0.972**	0.954
F1 Score	**0.838**	0.835
Training time (hr)	24.04	**7.41**
Exported model size (MB)	**4.12**	512

5 Discussion

The investigation found that both models are extremely effective in classifying whether or not patients are in pain (F1 scores above 0.830) and predicting actual PSPI scores (Pearson correlations above 0.950) as found in Table 1 in section 4.2. Also, Figure 3 and Figure 4 in section 4.1 shows that the training loss across the 5 epochs follows a similar trend suggesting that with limited hardware, both models would have acceptable levels of training MSE after training for at least 3 epochs.

The greatest difference between the two models is the running time, whereby the running time of the Main paper model is much longer than the ExtendedMTL4Pain model. This is because the pre-processing of the Main paper is more complicated, since images are taken from different angles, the Main paper normalizes different angles of input images after face detection and cropping. This extra step leads to a slightly more generic model as demonstrated by the Pearson correlation and the F1 scores. However, while the VGG16 architecture used in ExtendedMTL4Pain is a large architecture, it should be noted that this investigation only dealt with stage 1 out of 3. This means that with the entire architecture of ExtendedMTL4Pain, the training time can be expected to be longer since the original model could not train all three stages at once due to GPU limitation [15]. In addition to the training time, the Main paper uses pairwise training whereas the ExtendedMTL4Pain model uses pointwise training. Instead of mapping each image to a PSPI score, it learns the difference between the target image (PSPI>0) and the reference image (PSPI=0). In contrast, there are many cases where the PSPI score is 0 in the training and test sets of the ExtendedMTL4Pain model. Therefore, it is expected that the Main paper model has a better recall than the ExtendedMTL4Pain model in a real-world application, since generally, pointwise training performs worse than pairwise training [14].

While the models performed very well in predicting the F1 score and the Pearson correlation, the experiments had a few limitations. One of them is the lack of available powerful GPUs which means that conducting training at higher epochs is impossible given the time constraints. In fact, the Main paper mentions that their best results came from an epoch of 70 [11], which is an extremely huge difference compared to our 5. Also the ExtendedMTL4Pain paper uses 50 epochs to obtain their optimal results, which means that this model can also benefit from higher epochs since the training loss does not to show any signs of plateauing (Figure 3 and Figure 4). Also, another limitation is that our dataset does not have the action units (AU), video data and VAS of each image, which ExtendedMTL4Pain uses extensively in stage 2 and stage 3. This meant that the true potential of the second model can be explored further provided the Kaggle dataset had this type of data. Nevertheless, both models manage to precisely predict the PSPI scores despite the limitations which is of great use to a medical professional.

For medical professionals with limited data (i.e pictures with PSPI scores only), the model proposed by the Main paper is recommended. This is because the pairwise training method the Main paper employs results in better predictions of PSPI scores since it has higher F1 and Pearson correlation scores compared to the ExtendedMTL4Pain paper, which is important in predicting initial pain levels in a patient based on pure facial features. In addition, a medical professional will skip the expensive and time-consuming training of the Main paper (Table 1) and will use the smaller final trained model size so it is easier to store and is cheaper to use, which is very useful if there is a tight budget, making the Main paper superior compared to the ExtendedMTL4Pain paper.

6 Conclusion

In conclusion, for a medical professional with pictorial data with only annotated PSPI scores (i.e, limited dataset), this investigation shows that the Main paper model is more effective than the ExtendedMTL4Pain model, though both models are viable. Thanks to the pairwise training and smaller model size, the main paper model yields slightly better Pearson correlation and F1 scores during testing compared to the ExtendedMTl4Pain model. This makes the main paper model a more suitable candidate for real-world applications.

7 List of Contributions

Zhimao Lin: Implemented the training module and the processing of data module for both models, conducted final training and testing of both models using optimal hyperparameters, collected data and organized them in results section, helped to interpret strengths and weaknesses between models with respect to the results in the discussion, helped to connect experimental results to the context of medical practice, helped to interpret limitations on the experiments.

Christopher Demario Sastropranoto: Implementation of the testing metrics module, hyperparameter tuning of the Main paper, wrote Architectures of the Two Papers' section, wrote methods section, helped to connect experimental results to the context of medical practice, helped to interpret limitations on the experiments, helped to interpret strengths and weaknesses between models with respect to the results in the discussion.

Yasamin Zarghami: Helped with the implementation of the training module and processing of data, hyperparameter tuning of the ExtendedMTL4Pain paper, wrote introduction section, helped to interpret limitations on the experiments, helped to interpret strengths and weaknesses between models with respect to the results in the discussion, helped to connect experimental results to the context of medical practice, wrote conclusion and abstract.

References

[1] Pain recognition from facial expression dataset. `https://www.kaggle.com/datasets/coder98/emotionpain`.

[2] Tadas Baltrušaitis, Marwa Mahmoud, and Peter Robinson. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 06, pages 1–6, 2015.

[3] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230, 000 3d facial landmarks). *CoRR*, abs/1703.07332, 2017.

[4] Raymond Veldhuis Dan Zeng and Luuk Spreeuwers. Gradient-based learning applied to document recognition. pages 1–46, 1998.

[5] Raymond Veldhuis Dan Zeng and Luuk Spreeuwers. A survey of face recognition techniques under occlusion. page 3, 2020.

[6] Sebastian Kaltwang, Ognjen Rudovic, editor="Bebis George Pantic, Maja", Richard Boyle, Bahram Parvin, Darko Koracin, Charless Fowlkes, Sen Wang, Min-Hyung Choi, Stephan Mantler, Jürgen Schulze, Daniel Acevedo, Klaus Mueller, and Michael Papka. Continuous pain intensity estimation from facial expressions. In *Advances in Visual Computing*, pages 368–377, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[7] Michelle S. Bourgeois Kathryn M. Yorkston and Carolyn R. Baylor. Communication and aging. *National Library of Medicine*, 2011.

[8] Zakia Hammal Kenneth M. Prkachin. Computer mediated automatic detection of pain-related behavior: Prospect, progress, perils. *Frontiers in Pain Research*, 2021.

[9] Patrick Lucey, Jeffrey F. Cohn, Kenneth M. Prkachin, Patricia E. Solomon, and Iain Matthews. Painful data: The unbc-mcmaster shoulder pain expression archive database. In *2011 IEEE International Conference on Automatic Face Gesture Recognition (FG)*, pages 57–64, 2011.

[10] Andrew Zisserman Omkar M. Parkhi, Andrea Vedaldi. Deep face recognition. pages 1 – 5, 2015.

[11] Siavash Rezaei, Abhishek Moturu, Shun Zhao, Kenneth M. Prkachin, Thomas Hadjistavropoulos, and Babak Taati. Unobtrusive pain monitoring in older adults with dementia using pairwise and contrastive training. *IEEE Journal of Biomedical and Health Informatics*, 25(5):1450–1462, 2021.

[12] Pau Rodriguez, Guillem Cucurull, Jordi Gonzàlez, Josep M. Gonfaus, Kamal Nasrollahi, Thomas B. Moeslund, and F. Xavier Roca. Deep pain: Exploiting long short-term memory networks for facial expression classification. *IEEE Transactions on Cybernetics*, pages 1–11, 2017.

[13] Zhen Lei Hailin Shi Xiaobo Wang Stan Z. Li Shifeng Zhang, Xiangyu Zhu. S3fd: Single shot scale-invariant face detector. 2017.

[14] Bernd Frick Daniel Kaimann Eyke Hullermeier Vitalik Melnikov, Pritha Gupta. *Theoretical Foundations of Machine Learning Krakow*, 25:73 – 83, 2016.

[15] Xiaojing Xu, Jeannie S Huang, and Virginia R De Sa. Pain Evaluation in Video using Extended Multitask Learning from Multidimensional Measurements. In Adrian V. Dalca, Matthew B.A. McDermott, Emily Alsentzer, Samuel G. Finlayson, Michael Oberst, Fabian Falck, and Brett Beaulieu-Jones, editors, *Proceedings of the Machine Learning for Health NeurIPS Workshop*, volume 116 of *Proceedings of Machine Learning Research*, pages 141–154. PMLR, 13 Dec 2020.

A Appendix A

Table 2: Tuning of the drop_out hyper-paramter

drop_out	0	0.25	Percentage Change
Average Loss	5.178995132	15.0884738	65.67581849
Accuracy Score	0.2333333333	0	N/A
Pearson correlation	0.3710483845	0	N/A
F1 Score	0.6220095694	0	N/A

Table 3: Tuning of the fc2_size hyper-paramter

fc2_size	100	200	300	Percentage Change
Average Loss	3.225547671	5.703956842	5.428508401	43.45069992
Accuracy Score	0.2466666667	0.3	0.2533333333	21.62162162
Pearson correlation	0.6915981351	0.4794852984	0.5368247921	44.23761008
F1 Score	0.6705882353	0.6326530612	0.6629213483	5.996204934

Table 4: Tuning of the learning_rate hyper-paramter

learning_rate	0.01	0.001	0.0001	Percentage Change
Average Loss	4.394054294	4.067122459	2.421227574	44.89764094
Accuracy Score	0.2066666667	0.2533333333	0.2866666667	38.70967742
Pearson correlation	0.4840889572	0.5155533549	0.79398312	64.01595371
F1 Score	0.6444444444	0.6631578947	0.6918918919	7.362534949

Table 5: Tuning of the batch_size hyper-paramter

batch_size	50	100	150	200	Percentage Change
Average Loss	3.75862209	5.151902676	14.17789268	5.857748508	73.48955749
Accuracy Score	0.32	0.24	0	0.2066666667	54.83870968
Pearson correlation	0.7523603134	0.5230982204	nan	0.1923961556	291.0474776
F1 Score	0.7419354839	0.6161616162	0	0.5588235294	32.76740238

B Appendix B

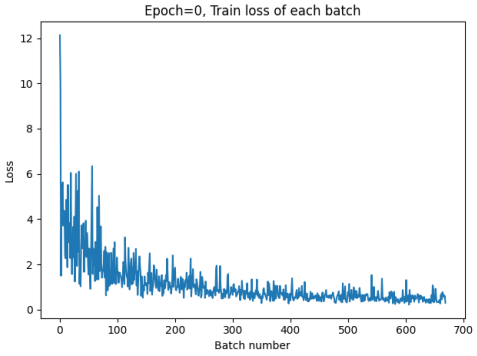

Figure 5: Main paper: The average MSE loss of each batch in Epoch 1.

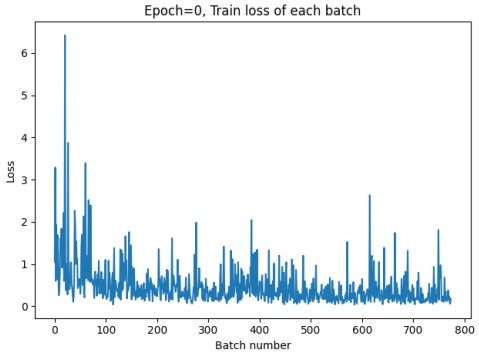

Figure 6: ExtendedMTL4Pain paper: The average MSE loss of each batch in Epoch 1.

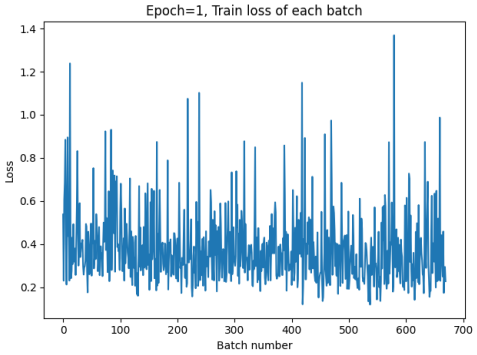

Figure 7: Main paper: The average MSE loss of each batch in Epoch 2.

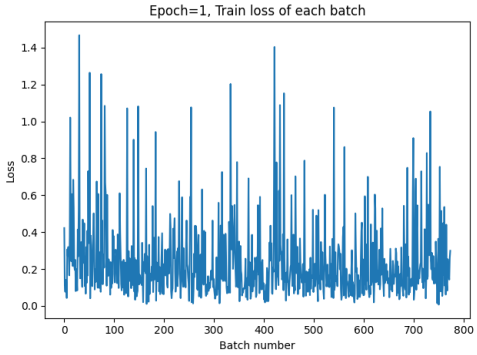

Figure 8: ExtendedMTL4Pain paper: The average MSE loss of each batch in Epoch 2.

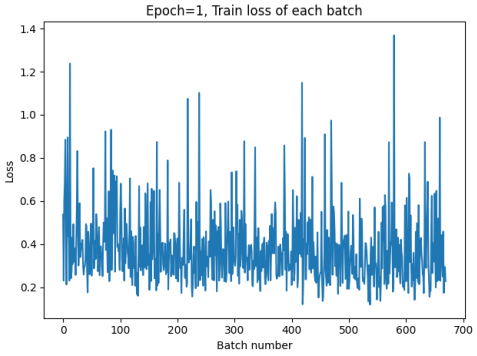

Figure 9: Main paper: The average MSE loss of each batch in Epoch 2.

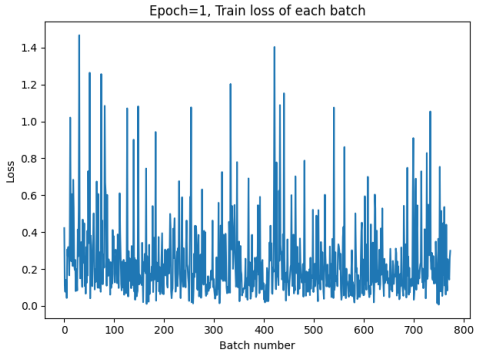

Figure 10: ExtendedMTL4Pain paper: The average MSE loss of each batch in Epoch 2.

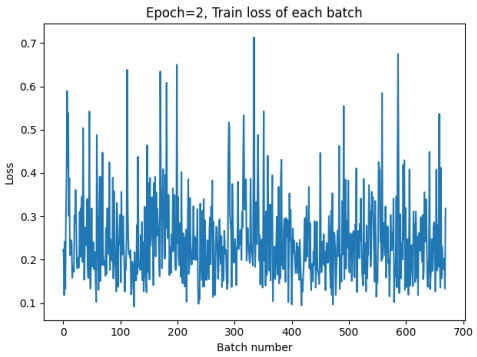

Figure 11: Main paper: The average MSE loss of each batch in Epoch 3.

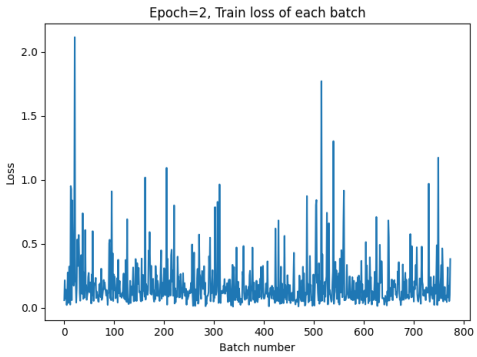

Figure 12: ExtendedMTL4Pain paper: The average MSE loss of each batch in Epoch 3.

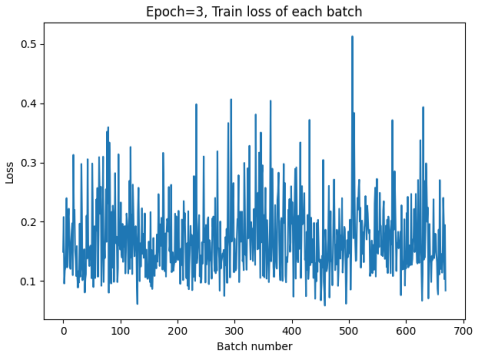

Figure 13: Main paper: The average MSE loss of each batch in Epoch 4.

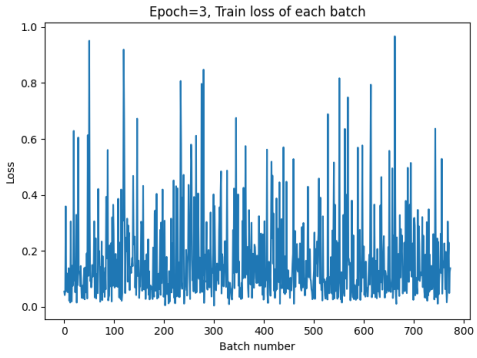

Figure 14: ExtendedMTL4Pain paper: The average MSE loss of each batch in Epoch 4.

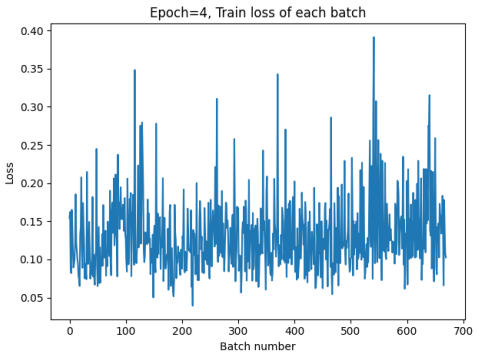

Figure 15: Main paper: The average MSE loss of each batch in Epoch 5.

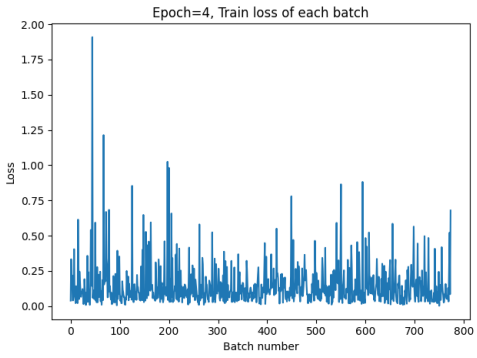

Figure 16: ExtendedMTL4Pain paper: The average MSE loss of each batch in Epoch 5.