2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020) | 978-1-7281-3079-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/FG47880.2020.00071

Three-level Training of Multi-Head Architecture for Pain Detection

Saandeep Aathreya Sidhapur Lakshminarayan, Saurabh Hinduja and Shaun Canavan
Computer Science and Engineering, University of South Florida, Tampa, Florida

***Abstract*— Precise pain detection is a complex task even for trained professionals. There are many occasions where self-reporting fails to capture the decisive pain measurement. Facial expressions help extract the subtle emotions which can be leveraged to detect pain levels. In this paper, we present our approach to task 1 of the FG 2020 EmoPain Challenge: Pain-related Behavior Analysis, as well as our experimental design and results. We utilize a multi-head approach with the combined features of facial action units, facial landmarks, HOG and deep features. This multimodal approach provides insight into the contribution of each of these features and their consolidated effect. To improve our regression model, we adopt a three-level architecture where we observe an increase in prediction accuracy as the levels deepen. We record results comparable to the baseline on the challenge validation set.**

I. Introduction

Facial expression recognition is an important field as it has many real-world applications such as driver frustration detection, assisting robots and pain detection in children and elderly. The need for pain detection primarily arises due to the inability of the patients to accurately self-report their pain or their incapacity to express pain [6]. Another roadblock might be the difference in understanding the pain scales between patient and a doctor. This deems pain as a subjective experience which does not have a direct metric for measurement. Consequentially, pain assessment becomes a highly influential task given the nuances in its nature. A well-grounded assessment of pain is necessary to identify a suitable anodyne. Many self-reporting and observational standards are used to evaluate pain intensities. Verbal representation, Numeric Scale Representation (NRS) [25] where patients are usually asked to rate their pain, Visual Analog Scales (VAS) [16] where patients point to a scale that lines up with their pain intensity, and pain dairies are some of the examples. However, Self-reporting cannot be taken into account when dealing with patients who are incapable of communicating the pain they experience. These may include newborns, autistic patients or bed-ridden elderly. In these cases, observational measurement needs to be performed by a mediator under a clinical settings (e.g., Behavioral pain scale (BPS) [28], Neonatal infant pain scale (NIPS) [8]). This requires any caregiver to constantly monitor the patient's health level in order to minimize the error in their labelling. This is certainly not feasible in a real-world scenario.

Facial expressions are an informative indicator of pain in behavioural scaling [15] and this has inspired research into automatically detecting pain levels from facial features. Current research uses a diverse set of features such as facial expression, body movement, and speech. Using the geometric facial features have been found useful in part due to vast information which can be extracted from them [15]. There are several approaches that make use of the features extracted from facial expressions [2] such as landmarks [9], facial action units [24] and hand-crafted deep features [6] to classify/predict the pain levels. Egede et. al [6] makes use of combination of hand-crafted and deep learned features extracted from 66 facial landmarks and the action units which achieved a Root Mean Squared Error of 0.99. Kaltwang et. al [10] incorporated facial action units and body movement and used multiple datasets in their experiments and achieved an RMSE of 1.69. Roy et. al [20] make use of an Active Appearance Model and Support Vector Machine to detect pain levels up to 4 levels with an accuracy of 82% on the UNBC McMaster pain database [15]. UNBC McMaster contains 17.29% of the frames labeled as pain, which can make it difficult to learn the pain labels [6].

Motivated by these works, we propose a three-level, multi-head deep architecture for task 1 of the FG 2020 EmoPain Challenge [5]: Pain-related Behavior Analysis. The main contributions are 3-fold, and can be summarized as follows.

1) We propose a multi-head deep architecture that uses three level of training, to detect pain.
2) An ablation study is performed to evaluate the impact of each level, of the proposed architecture, on the accuracy of detecting pain.
3) Proposed architecture is evaluated using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Pearson Correlation Coefficient (PCC), and Concordance Correlation Coefficient (CCC). We report results comparable to the FG 2020 EmoPain baseline results.

II. Three-level Multi-Head Architecture

A. Motivation for Architecture

We propose a multi-head approach [17], for detecting pain, that undergoes three levels of training. Chu Y et. al [3] explains the strong behaviour of multiple feature set in detecting pain. K. Liu et. al [14] explains the overfitting nature of weak features. Motivated by this, we propose to use multiple feature sets for training, where each head outputs a vector of deep features, which are then fused together. We have chosen this architecture as multi-head architectures use relevant information, at each level, to find the correlations between the different data types [11].

A single-level model has to go through a higher number of iterations to identify the best hyper parameters to produce a low cost error. Additionally, a hierarchical structure allows for definitive outflow of information between the levels. This is a computationally cheaper way of regression especially

978-1-7281-3079-8/20/$31.00 ©2020 IEEE
DOI 10.1109/FG47880.2020.00071

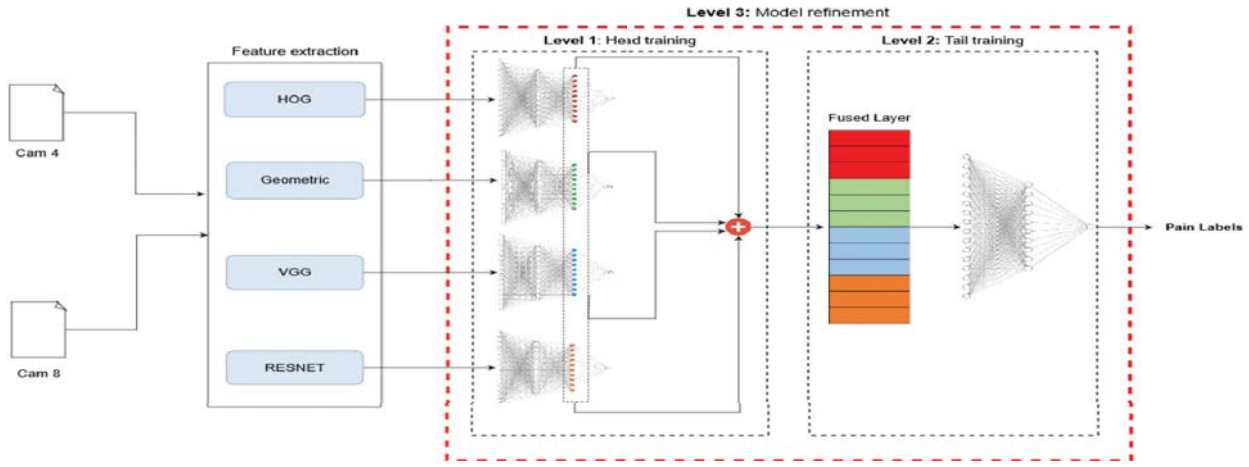

Fig. 1: Multi-head three-layer architecture. Level 1 trains individual feed-forward neural networks for each feature type (i.e. multi-head). Deep features are then extracted from each network and fused for input to level 2 network. At level 2, network from level 1 is frozen, and 3 new layers are added, which are trained on fused feature vector (each feature type is represented by a colored layer, 3 parts are a simplified representation of the extracted features) (i.e. tail). Level 3 unfreezes all layers, and retrains entire model (i.e. level 1 and 2) for model refinement.

with the inclusion of multimodal features [21], [23]. Multi-level modelling can also improve the interaction between models. Given levels $L1, L2$, and $L3$, we can see the communication between them allowing us to interpret the results. At each level, the confidence of the output by the corresponding model increases due to the hierarchical form which makes the predictions statistically trustworthy [17], [21]. Another benefit is the convergence rate of the multi-level model can be faster than a single-level. Our proposed architecture combines the benefits of a multi-level architecture with the positive impact that multimodal classification can have on pain detection accuracy [27]. Motivated by the work of Monwar et. al [18], where feed-forward neural networks were trained on facial features, and Minetto et al. [17], that used a multi-head network for geospatial land classification, we create a multi-head architecture, with a lateral division where multiple features are used as input to a regression-based feed-forward neural network.

B. Level Hierarchy

1) Level 1: Here, we train the multi-head level of our architecture. Each head is a feed-forward neural network consisting of one input layer, one output layer and 2 hidden layers, where the subsequent hidden layers have $\frac{\text{No of neurons in previous layer}}{2}$ neurons. The final hidden layer was kept at 50 nodes to keep it consistent for further fusion (i.e. output of 50 features). A batch size of 250 with 250 epochs were used. Dropout of 20% was added after each layer to avoid over-fitting. At this level, the network is trained for each feature (e.g. HOG features), using Adam optimizer [12] with a learning rate of 0.01. We use RMSE as our loss function, and the error is measured by MAE. For our experiments, we train 4 networks, 1 for each feature type we use (HOG, Geometric, VGG [13], and RESNET [13]). From each network, we extract a feature vector of size 50, from each of the fully connected layers. We then create a new feature vector, f, of size 200 (4×50), by concatenating each vector,

$$f = [Geometric_{HL3}, VGG_{HL3}, HOG_{HL3}, RESNET_{HL3}], \quad (1)$$

where $Geometric_{HL3}$, VGG_{HL3}, HOG_{HL3}, and $RESNET_{HL3}$ are the deep features of the geometric, VGG, HOG, and RESNET networks, respectively. This new vector, f is then used as input to level 2.

2) Level 2: Here, we train the tail of our proposed architecture, which consists of the fusion layer. It has been shown that fusion takes advantage of the underlying multi-modal features and improves the regression abilities of the model [21]. Considering this, we use the fused feature vector f, from level 1, as input to the network at this level. To construct this level, we first freeze the network layers from level 1, and add 3 new layers (1 input, 1 hidden, and 1 output layer). These layers are trained, for 100 epochs with a batch size of 250, and learning rate of 0.01. Using this approach allows for faster convergence of the network at this level [23]. The final network weights are then saved for level 3.

3) Level 3: We consider the final level to be the model refinement level. Here, the entire architecture (i.e. level 1 *and* level 2) is retrained. The difference being, the weights of the head are initialized to the final weights of the heads at level 1 training and the weights of the tail are the final weights of the tail at level 2 training. At this level, we unfreeze the layers that were frozen in level 2 (i.e. level 1 layers), and they are trained with a learning rate of 0.001. The output of this network is the 11-point scale frame-wise pain intensity labels ($[0, 10]$) with linear activation. As we will show in Section III-B.4, this model refinement results in a positive impact to detecting pain. See Fig. 1 for an overview of the

proposed architecture.

III. Experimental design and Results

A. Dataset

In our experiments, we have used the state-of-the-art pain related dataset EMOPain [1]. It is a multimodal, fully labelled dataset which has high-resolution face videos captured from multiple cameras positioned at different locations. In addition to facial data, full body 3D motion capture and the electromyographic signals from the back muscles have been captured. For our experiments, we utilize the features extracted from the face videos. We use the following features to conduct experiments and test our architecture- *Geometric features* which are composed of headpose, 2D/3D landmarks, Facial Action Unit [7] occurrence and Facial Action Unit intensity, and *HOG* features [4]. In addition to this, two sets of deep features have been collected namely *RESNET* [26] and *VGG* [22].

This dataset consists of 50 participants split between 21 male and 29 female. Out of these, 22 participants (7 male and 15 female) were chronic lower back patients (CLBP) and 28 participants (14 male and 14 female) were termed healthy patients. The mean age of CLBP patients was 50.5 while that of healthy patients was 37.1. Subjects were labelled per frame on a 11-point scale and the average of the all the tasks performed (Normal or Difficult) was considered the final label. The features were extracted from two cameras (camera 4 and camera 8). In our experiments, we have utilized both the camera features by selecting only the informative frames (see Section III-B.1 for details). The training data comprised of 19 subjects (11 healthy and 9 CLBP) and the validation data comprised of 9 subjects (6 healthy and 3 CLBP).

B. Experiments and Results: Ablation Study

To validate our proposed architecture, we conducted 3 experiments to learn how each level impacts the accuracy of detection. (1) Experiment 1: we evaluate the accuracy of each individual network (i.e. single modality feature vector); (2) Experiment 2: the accuracy of the fused features in the level 2 network are evaluated; (3) Experiment 3: final accuracy of three-level multi-head architecture is evaluated. For each experiment, we use the following evaluation metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Pearson Correlation Coefficient (PCC), and Concordance Correlation Coefficient (CCC).

MAE is the average of the absolute difference between the predictions and ground truth and is given by

$$MAE = \frac{\sum_{i=1}^{N} |(y_{true} - y_{pred})|}{N}, \tag{2}$$

where N is the total number of samples, y_{true} is the ground truth, and y_{pred} is the predicted value. *RMSE* tells us how spread out are the data from the predicted line. It is a measure of the difference between the predicted value and ground truth. RMSE values are mainly used in regression inspection to evaluate a model, and is given by

$$RMSE = \frac{\sum_{i=1}^{N} \sqrt{(y_{true} - y_{pred})^2}}{N}, \tag{3}$$

where N is the total number of samples, y_{true} is the ground truth, and y_{pred} is the predicted value. The main difference between RMSE and MAE is taking the square in RMSE results in higher weight to a larger error.

PCC measures the linear correlation between two variables y_{pred} and y_{true} as

$$PCC = \frac{cov(y_{true}, y_{pred})}{\sigma_{y_{true}} \sigma_{y_{pred}}}, \tag{4}$$

where cov is the covariance between the y_{true} and y_{pred}, $\sigma_{y_{true}}$ and $\sigma_{y_{pred}}$ are the standard deviation of y_{true} and y_{pred} respectively. PCC greater than 0 implies a positive correlation between the two variables and a PCC less than 0 implies negative correlation. A PCC of 0 implies there is no linear correlation between the two variables.

CCC measures the agreement between two variables as

$$CCC = \frac{2\rho_{y_{true}} \sigma_{y_{pred}}}{\sigma_{y_{true}}^2 + \sigma_{y_{pred}}^2 + (\mu_{y_{true}} + \mu_{y_{pred}})^2}, \tag{5}$$

where ρ defines the correlation between the two variables, $\mu_{y_{true}}$ and $\mu_{y_{pred}}$ are the means of the two variables and $\sigma_{y_{true}}$ and $\sigma_{y_{pred}}$ are the standard deviation.

1) Data Integrity: We first investigated the data to determine the frame-level details which added no value to the training and deteriorated them (i.e. some of the training data is inconsistently labeled). One of the main properties which will give a clue about this is the head pose coordinates. For some of the files, the head pose features had invalid values in them (e.g. 10000 as head pose value, 9 digit landmark coordinates). We have captured these details at the frame-level and have removed them from our training set. We mainly focus on the geometric feature vector and observe the head pose columns to filter out the invalid rows. We notice that for these rows, the remaining features such as action unit occurrence and action unit intensities are zero. We remove the corresponding rows in the remaining feature set to maintain compatibility with our models. For testing purposes, to ensure the final model evaluation is compatible with the ground truth test labels (for calculation of RMSE, MAE etc), our predictions per subject will match the pain labels frames of the subject. We adopted the following rules to fill missing labels from invalid frames.

- For any continuous missing frames, obtain the predicted label of the frames before and after the invalid values and take the weighted average of the predictions. Fill the missing labels with the new weighted average.
- For frames which have been removed from the end, fill the rows with the last valid frame prediction.

We refer to the inconsistent data as *iData (invalid data)*, and *vData (valid data)*. As can be seen in Table I, the invalid data has a negative impact on detecting pain. One of the reasons for invalid frames can be attributed to the exercise being performed. For example, for exercises that requires the

TABLE I: Pain detection results, from each level of proposed architecture, on the EmoPain [1] validation and test sets.

Phase	Level	Model	RMSE		MAE		PCC		CCC	
			vData	iData	vData	iData	vData	iData	vData	iData
Validation	Level 1	HOG	1.8680	1.8840	1.0030	1.0300	-0.0680	-0.0620	-0.0470	-0.0050
		Geometric	1.8220	1.8480	0.7970	0.8230	-0.0120	0.0560	-0.0070	-0.0060
		RESNET	1.8840	1.8620	0.8790	0.8890	-0.0400	-0.0400	-0.0300	-0.0300
		VGG	1.7970	1.8800	0.8850	0.8960	0.0500	0.0300	0.0300	0.0300
	Level 2	Fusion (All 4 features)	1.8610	1.8800	1.0900	1.2300	-0.0670	-0.0690	-0.0540	-0.0520
		Fusion (Geo + VGG)	**1.7800**	1.8400	**0.7550**	1.2400	-0.0400	0.0020	-0.0200	-0.0300
	Level 3	Retrain (All 4 features)	1.7800	1.8700	0.8000	0.8600	-0.0090	-0.0010	-0.0310	-0.0310
		Retrain (Geo + VGG)	**1.6740**	1.7900	**0.7320**	0.7500	-0.0090	-0.0100	-0.0010	-0.0020
	Baseline [5]	Fusion	1.6900		1.2600		0.2500		0.1800	
Test	Fusion		5.4800		0.8550		0.0034		0.0240	
	Baseline [5]		1.4100		0.9100		0.1000		0.0600	

patient to bend forward, only camera 8 data is active. The data present in camera 4 can be considered invalid. Another reason might be the data simply wasn't captured for some of the frames. Since we are considering both the cameras for our prediction, we are removing the invalid frames from both the camera 4 and camera 8 to maintain consistency across the cameras. For the 3 experiments, detailed below, the invalid data is removed from training.

2) Experiment 1: We evaluated the accuracy of the individual networks of level 1. As detailed in Section II-B.1, the loss function was RMSE and the network accuracy was monitored as MAE for training our networks, however, for the challenge, we also calculated PCC, and CCC scores from the results of each network (Table I). VGG features had the lowest RMSE, and highest PCC and CC scores with 1.797, 0.05, and 0.03 respectively. For MAE, geometric features had the lowest error with 0.797. While these features were the best for the individual metrics, all features had relatively stable performance. The standard deviation, across the four features, for RMSE, MAE, PCC, and CCC were 0.04, 0.08, 0.05, and 0.03, respectively. This shows a small amount of variation in accuracy between each of the features.

3) Experiment 2: In experiment 1, we wanted to evaluate the accuracy of each individual feature type. Here, in experiment 2, we analyze our multimodal approach (i.e. fusion) to detecting pain. Considering this, we calculate the RMSE, MAE, PCC, and CCC scores for the fusion of all modalities (HOG, Geometric, RESNET, and VGG), which are 1.861, 1.09, -0.067, and -0.054, respectively. For all evaluation metrics, the fusion of all of the modalities resulted in decreased performance. Considering this, we also evaluated the top features for level 1, which were Geometric (MAE), and VGG (RMSE, PCC, and CCC). To evaluate these features, we performed a separate fusion of just those deep features (i.e. from the output of level 1), and subsequently trained level 2 on these fused features. This resulted in an RMSE, MAE, PCC, and CCC of 1.78, 0.755, -0.04, and -0.02, respectively. This resulted in an overall higher PCC and CCC score compared to level 2, and lower RMSE and MAE error compared to both level 1, and the fusion of all features at level 2. These results can partially be explained by using RMSE as the loss function and evaluating the accuracy of our networks with MAE.

4) Experiment 3: The final experiment was done to evaluate the impact of model refinement (i.e. level 3). Here we conduct the same experiments as done for level 2 (fusion of all features, as well as fusion of Geometric and VGG features). When fusing all features, the model refinement level results in an RMSE, MAE, PCC, and CCC of 1.78, 0.8, -0.009, and -0.031, respectively. Similar, to level 2 experiments, fusing only geometric and VGG features resulted in a positive impact for RMSE, MAE, and CCC with 1.674, 0.732, and -0.001, respectively. For the CCC score, model refinement resulted in the same score as level 2 (-0.009). At level 1, an average RMSE of 1.843 was achieved across all features types. With model refinement at level 3, RMSE was decreased by 0.169, showing improved accuracy for detecting pain. Similar positive refinement can be seen for the other evaluation metrics compard to the average of the four features. MAE resulted in a decrease of error by 0.159; CCC scores and PCC scores increased by 0.0085 and 0.0125, respectively. These results are encouraging, improving upon the baseline results for RMSE and MAE, and showing comparable PCC and CCC scores (Table I).

IV. Conclusion and Future work

In this paper, we proposed a multi-head, multi-level architecture for task 1 of the FG 2020 EmoPain challenge. The proposed architecture incorporates multimodal data, that captures the underlying details of the features, for predicting the pain on an 11 point scale. We have validated the proposed architecture by performing an ablation study showing that the multimodal approach outperforms a single modality and network. We facilitated this evaluation by comparing the RMSE, MAE, PCC, and CCC values of individual models against the proposed architecture and observed lower error values and higher correlations on average. We have shown that the RMSE and MAE values of our model outperform the baseline, and the PCC and CCC scores are comparable. As part of future work, we will utilize camera 4 and camera 8 as separate models and apply an ensemble approach [19] on both. Each head can provide a probability of the output which can be the confidence of the output.

Acknowledgment

This material is based on work that was supported in part by an Amazon Machine Learning Research Award.

REFERENCES

[1] M. S. H. Aung, S. Kaltwang, B. Romera-Paredes, B. Martinez, A. Singh, M. Cella, M. Valstar, H. Meng, A. Kemp, M. Shafizadeh, A. C. Elkins, N. Kanakam, A. de Rothschild, N. Tyler, P. J. Watson, A. C. d. C. Williams, M. Pantic, and N. Bianchi-Berthouze. The automatic detection of chronic pain-related expression: Requirements, challenges and the multimodal emopain dataset. *IEEE Transactions on Affective Computing*, 7(4):435–451, Oct 2016.

[2] T. Baltrušaitis, P. Robinson, and L. Morency. Openface: An open source facial behavior analysis toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10, March 2016.

[3] H. J. S. Y. Chu Y, Zhao X. Physiological signal-based method for measurement of pain intensity. 2017.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005.

[5] J. Egede, T. Olugbade, C. Wang, S. Song, N. Berthouze, M. Valstar, A. Williams, H. Meng, M. Aung, and N. Lane. Emopain challenge 2020: Multimodal pain evaluation from facial and bodily expressions. *arXiv preprint arXiv:2001.07739*, 2020.

[6] J. Egede, M. Valstar, and B. Martinez. Fusing deep learned and hand-crafted features of appearance, shape, and dynamics for automatic pain estimation. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pages 689–696, May 2017.

[7] P. Ekman. What the face reveals: Basic and applied studies of spontaneous expression using the facial action coding system (facs). *Oxford University Press*, 1997.

[8] A.-M. Gallo. The fifth vital sign: Implementation of the neonatal infant pain scale. *Journal of Obstetric, Gynecologic, & Neonatal Nursing*, 32(2):199–206, 2003.

[9] S. Kaltwang, O. Rudovic, and M. Pantic. Continuous pain intensity estimation from facial expressions. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, C. Fowlkes, S. Wang, M.-H. Choi, S. Mantler, J. Schulze, D. Acevedo, K. Mueller, and M. Papka, editors, *Advances in Visual Computing*, pages 368–377, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[10] S. Kaltwang, S. Todorovic, and M. Pantic. Doubly sparse relevance vector machine for continuous facial behavior estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1748–1761, Sep. 2016.

[11] S. Kaushik, A. Choudhury, N. Dasgupta, S. Natarajan, L. Pickett, and V. Dutt. *Ensemble of Multi-headed Machine Learning Architectures for Time-series Forecasting of Healthcare Expenditures*, page In press. 10 2019.

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] D. Kollias, P. Tzirakis, M. A. Nicolaou, A. Papaioannou, G. Zhao, B. Schuller, I. Kotsia, and S. Zafeiriou. Deep affect prediction in-the-wild: Aff-wild database and challenge, deep architectures, and beyond. *International Journal of Computer Vision*, 127(6-7):907–929, Feb 2019.

[14] K. Liu, Y. Li, N. Xu, and P. Natarajan. Learn to combine modalities in multimodal deep learning, 2018.

[15] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, and I. Matthews. Painful data: The unbc-mcmaster shoulder pain expression archive database. In *Face and Gesture 2011*, pages 57–64. IEEE, 2011.

[16] M. Lynch. Pain as the fifth vital sign. *Journal of intravenous nursing : the official publication of the Intravenous Nurses Society*, 24:85–94, 03 2001.

[17] R. Minetto, M. P. Segundo, and S. Sarkar. Hydra: An ensemble of convolutional neural networks for geospatial land classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):6530–6541, 2019.

[18] M. M. Monwar and S. Rezaei. Pain recognition using artificial neural network. In *2006 IEEE International Symposium on Signal Processing and Information Technology*, pages 28–33, Aug 2006.

[19] R. Paul, M. Schabath, R. Gillies, L. Hall, and D. Goldgof. Mitigating adversarial attacks on medical image understanding systems. In *ISBI*, 2020.

[20] S. D. Roy, M. K. Bhowmik, P. Saha, and A. K. Ghosh. An approach for automatic pain detection through facial expression. *Procedia Computer Science*, 84:99–106.

[21] C. Scott and E. Mjolsness. Multilevel artificial neural network training for spatially correlated learning. 06 2018.

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[23] G. Szabó, J. Szüle, Z. Turányi, and G. Pongrácz. Multi-level machine learning traffic classification system. 02 2012.

[24] Y. . Tian, T. Kanade, and J. F. Cohn. Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):97–115, Feb 2001.

[25] A. Williams, H. Davies, and Y. Chadury. Simple pain rating scales hide complex idiosyncratic meaning. *Pain*, 85:457–63, 05 2000.

[26] S. Zagoruyko and N. Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016.

[27] G. Zamzmi, C.-Y. Pai, D. Goldgof, R. Kasturi, T. Ashmeade, and Y. Sun. An approach for automated multimodal analysis of infants' pain. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 4148–4153. IEEE, 2016.

[28] S. M. Zwakhalen, J. P. Hamers, and M. P. Berger. Improving the clinical usefulness of a behavioural pain scale for older people with dementia. *Journal of Advanced Nursing*, 58(5):493–502, 2007.