Deep Action Unit Classification using a Binned Intensity Loss and Semantic Context Model

Edward Kim
Department of Computing Sciences
Villanova University
Villanova, Pennsylvania 19085
Email: edward.kim@villanova.edu

Shruthika Vangala
Department of Computing Sciences
Villanova University
Villanova, Pennsylvania 19085
Email: svagal1@villanova.edu

***Abstract*—One of the most important cues for human communication is the interpretation of facial expressions. We present a novel computer vision approach for Action Unit (AU) recognition based upon a deep learning framework combined with a semantic context model. We introduce a new convolutional neural network training loss specific to AU intensity that utilizes a binned cross entropy method to fine-tune an existing network. We demonstrate that this loss can be more effectively trained in comparison to an L2 regression or naive cross entropy approach. The results of our binned cross entropy neural network are then passed to our semantic model, which utilizes the co-occurrence of action units for improved binary and real valued classification. Through our qualitative and quantitative results, we demonstrate the improvement of our framework over the current state-of-the-art.**

I. Introduction

In 1977, Albert Mehrabian postulated that our words (the literal meaning) accounts for only 7% of our overall message [1], and in fact most of our communication spectrum resides in the nonverbal space. Our body language, facial expressions, and tone all contribute more to human communication than the literal meaning of our words. In order to design more intelligent and responsive computers, machines will need to learn how to observe human behavior and infer their intent and emotional state through their interactions and facial expressions. Computer vision research has been using the Facial Action Coding System (FACS) to quantitatively categorize facial activity [2]. By recognizing the Action Units (AUs) that correspond to facial muscle motion, one can develop a system to recognize the emotion of a user through a combination of these AUs, see Figure 1 for several examples. Images encoded with AUs typically contain a label for the existence of a certain AU, as well as an intensity label that ranges from A-E, where "A" indicates the weakest trace of the presence of an AU and "E" is the maximum intensity. Computers can be trained to classify these AUs and intensity values through machine learning techniques.

In the past several years, a new machine learning technique has shown promising results in modeling complex computer vision systems e.g. deep convolutional neural networks (CNNs) [3], [4]. These networks have shown impressive improvements over existing methods when trained by massively multicore GPUs on millions of generic images [5]. However, in our scenario, we do not have access to millions of AU labels and intensity measurements since obtaining these labels are both extremely time consuming and require specific observer training. In our work, we describe a new AU classification approach for utilizing semantic CNNs in smaller scale classification tasks, while simultaneously maintaining the benefits of the large training data corpus. Our method is unique in FACS classification and our contributions are two fold:

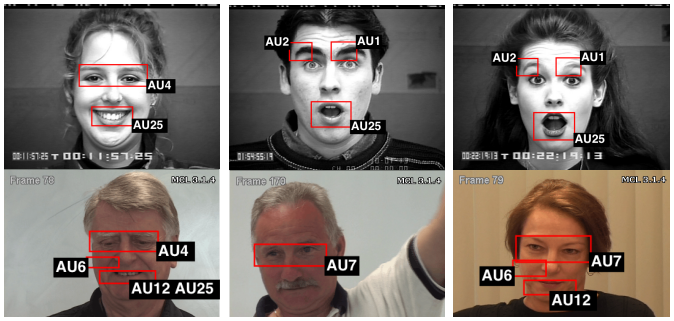


Fig. 1: Example faces from the CK+ database [6], [7] (row 1) and the UNBC-McMaster Pain archive [8], [9] (row 2) with a subset of activated AUs visualized. The approximate AU location is represented by a red bounding box determined by facial keypoints. ©Jeffrey Cohn

- We transform the issue of solving a real valued multi-AU intensity L2 regression problem to a binned cross entropy loss that can be modeled more efficiently and effectively by CNNs.
- We demonstrate that the classification task involving the presence and intensity of a specific AU can be obtained from our combined semantic conditional random field CNN model, and show the quantitative benefits of utilizing our multi-label, co-occurrence approach.

We present our experiments on two standard datasets and show notable improvement over other machine learning algorithms that utilize state-of-the-art image features.

II. Background and Related Work

Our work explores facial action unit recognition with machine learning using neural networks combined with co-occurrence relations. In this section, we will briefly review some of the literature from both domains, with a focus on the state-of-the-art and highly relevant prior works.

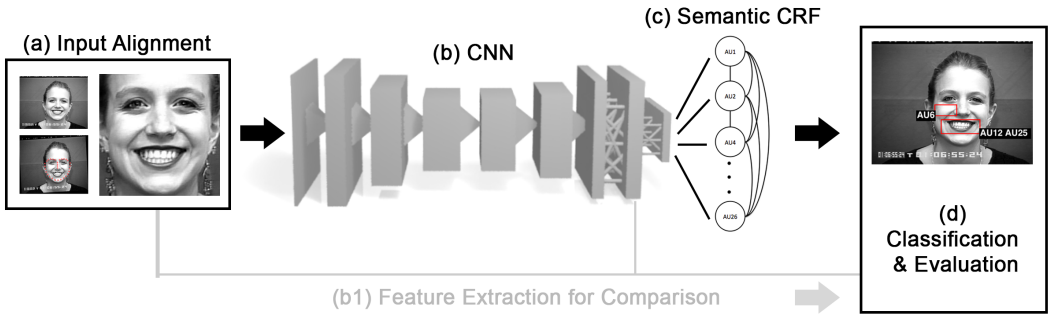


Fig. 2: Illustration of the major steps in our classification process (a)-(d). In (a), the input image is put through an alignment process to crop the face. Then in (b) we utilize a convolutional neural network with a binned intensity loss to provide the unary potentials in a semantic CRF (c). The fully connected edge potentials are generated from the training data and used in the final classification process (d). For experimental and validation purposes, we perform feature extraction at different steps in the process and use these in different classification methods, illustrated by (b1).

Facial expression through FACS classification - Facial expression analysis using facial muscle action units has been a popular approach [10]. Minimum Average Correlation Energy (MACE) [11] is an earlier work that uses an energy minimization technique to facilitate target detection which we use as a baseline for our comparisons. More recently, SVMs [12] and Bayes Nets [13], [14] are classification techniques used for AU recognition. Chew et al. [12] used Modified Correlation Filters (MDF) in an SVM framework that uses only a single hyperplane. Kapoor et al. [13] used a Bayesian Compressed Sensing (BCS) approach that models the sparsity in the label space to reduce the multiclass problem to a simpler regression task. Song et al. [14] built upon this work to explicitly utilize the co-occurrence of labels to boost the classification performance in a framework called BGCS. Our method utilizes a more complex machine learning framework using neural networks and co-occurrence potentials that has several benefits over these existing methods. Our framework naturally encodes multiple labels for shared AU recognition and it does not use hand-crafted image features. Rather, the relevant image features are automatically encoded through the hidden layers of the network and refined through a probabilistic model that resembles Rabinovich et al. [15].

Deep learning and Convolutional Neural Networks - Deep learning is a type of machine learning that uses an artificial neural network with multiple hidden layers of units between the input and output layers. Convolutional networks are distinct in that they use a convolutional filter layer that can process the 2D structure of images. These deep models, trained on large scale image data sources [5] have outperformed all other known methods in large scale challenges. AlexNet [3] and CaffeNet [4] are two examples of high performing neural networks with similar configurations. They contain seven layers with an input layer, five convolutional layers, two fully connected layers, and an output layer. The total number of parameters of the network is 60 million. Our work builds upon the architecture of CaffeNet for AU classification. Others have used shallow neural networks [16] for FACS classification, but lack the modeling complexity of deeper architectures. Recently, a seven layer deep neural network [17] was created for FACS recognition using a mean squared error (MSE) loss trained on a relatively small set of images. However, because of the small dataset, they overfit their network and in the following sections, we additionally show that their euclidean based loss has training difficulties and is less robust to outliers. In contrast, our method introduces a Binned Cross Entropy Loss, and we demonstrate that this proposed method can alleviate these problems.

III. Methodology

For our AU recognition framework, we first create a common alignment for all the images. Then, we estimate any missing data from the dataset, and train our neural network. We include in the methodology a feature extraction step for experimental comparison. We perform AU recognition using a custom binned loss function combined with a conditional random field. Each step is illustrated in Figure 2 and described in detail below.

A. Dataset Alignment

The first step of our framework is to create a common alignment for all the faces so we can more precisely train a classifier to detect the variations between expressions. We utilize two standard face datasets, the extended Cohn-Kanade (CK+) face database [6], [7] and the UNBC-McMaster Pain archive [8], [9]. Both of these datasets provide 60+ facial landmarks that correspond to the facial interest points.

Extended Cohn-Kanade (CK+) - The extended Cohn-Kanade (CK+) dataset consists of 593 recordings 123 subjects. The image sequences vary between 10 and 60 frames, but only the last frame (peak expression) has been FACS coded. The resulting size of the database is 593 frames with AU labels and intensity. The range of emotions expressed through this database cover a wide spectrum, including happiness, sadness, surprise, disgust, etc. Any missing intensity values in this dataset are regression imputed by our framework.

UNBC-McMaster (Pain AU) - The UNBC-McMaster Shoulder Pain Archive (Pain AU) is one of the largest databases of AU coded videos of spontaneous facial expressions. There are 200 sequences across 25 subjects and all frames are FACS encoded with intensity labels. The spontaneous nature of the video, even with control over the lighting and camera position, make this a very challenging dataset.

(a) Original image (b) Affine warped (c) Cropped

Fig. 3: Visualization of the image alignment steps taken to normalize the data. The original image (a) is first affine warped to a base shape shown by the face keypoints in (b). The 68 keypoints are shown, where only the first 31 (displayed in red) are used for the affine warp. The image is then cropped and contrast stretched for the final face (c).

Alignment - Given a 68 keypoint model of the input face obtained through manual or automatic means, we compute a mean face shape across the entire dataset and align each individual face to the mean shape through an affine warp of the outer 27 keypoints of the face and 4 keypoints of the nose bridge, see Figure 3 for details. We did not use a full 68 keypoint warp, as it could distort the facial muscle activations. A contrast stretch is then applied to the image, and the resulting face is used for the feature extraction step. Each image is represented in RGB space and resized to 227x227 pixels. Grayscale images are augmented to RGB color by repeating the intensity channel over each color channel.

B. Feature Extraction (for Comparison)

One of the key benefits of deep architectures is the automatic feature extraction capability, where no hand-crafted features need to be computed from an image. However, for evaluation purposes, we compare against two feature extraction methods that have shown good success in the literature.

PHOG features - The first descriptor we extract is a pyramid histogram of oriented gradients, PHOG [18]. The PHOG descriptor represents the local image shape and its spatial layout. To extract the PHOG descriptors from an image or image region, we first compute the gradient response using a sobel edge filter. If we use an 8 bin orientation histogram over 4 levels, the total vector size of our PHOG descriptor for each image is 680 bins.

CNN codes - CNN codes image features extracted from a convolutional neural network output on an internal layer of the network. Our approach is to remove the last fully connected classification or regression layer and use the 4096 dimensional vector activation outputs as the image feature. These activations are thresholded at zero by a ReLU activation function and are obtained for both training and testing in a separate classifier (like a Linear SVM). These codes have shown extraordinary potential for transfer learning on novel tasks [4].

C. Convolutional Neural Network

Building a convolutional neural network is a data intensive and time consuming task. For example, CaffeNet [4], the seven layer architecture that we utilize for our task, takes several weeks to train on the ImageNet [5] database, even using multiple GPUs. Since our deep neural network contains 60 million weight parameters, it is essential to have enough data to train the network without overfitting. But in our scenario, we only have hundreds or thousands of labeled faces. To still effectively use deep learning architectures, we turn to a different approach called "fine-tuning" [19]. Fine-tuning a network involves taking an existing deep neural network that has been pre-trained using millions of images and initializing our new network parameters with those given weights. We then replace the last fully connected layer with a new, uninitialized parameter layer and modify the number of outputs to 10, which is equal to recognizing the specified AUs in our classification task. The other layer weights from layer 1-6 in the network remain virtually stable from the previous training on ImageNet. Slight modifications of the weights in the early layers are enabled by a small fractional learning rate of $0.01\times$ base learning rate. The other parameters for fine-tuning the network are: base learning rate = 0.001, gamma = 0.1, momentum = 0.9, weight decay = 0.0005, over 2000 iterations. Our final architecture is an seven layer network with a 227x227x3 input layer, five convolutional layers, two fully connected layers, and a 10 node output layer. The internal layer sizes and filters mimic CaffeNet. With this configuration, and a mini-batch size of 200 samples, we can fine-tune a network in 40 minutes on a Tesla K40 GPU.

D. Data Augmentation - Intensity Imputation

In certain scenarios, we are given only partial data necessary for training our model. For example, in the CK+ dataset, the binary labels for the various AUs exist; however, in 78% of the cases, their intensity levels are partially labeled or not provided. We could simply ignore this partial information, but noticed a drop of about 3% on our final binary evaluation metrics. It is especially important to utilize all the data that we can when using data-hungry machine learning methods. Therefore, to incorporate this information, we need a principled method of filling in the absent data, i.e. multiple linear regression imputation.

Given the full data samples that have both AU and intensity labels, we can perform a multilinear regression, and based upon the predictors, compute the coefficients of the AU variables. We describe in our experiments that we isolate the dimensionality of our AU set to 10. These 10 AUs become our predictors, and the responses are the intensity of the AUs. For a set of labeled AUs in the dataset that do not have intensity values, we can now estimate the floating point values through multiple (3) regression imputations and average the result. Because our framework requires integer intensities, we round to the nearest intensity number ($<= 5$) and use this as our final imputed data.

E. Loss Function

One of the most critical pieces of any machine learning algorithm is choosing the correct loss function. In the AU classification task, there can be more than one positive output label e.g. AU 1, 2, and 25 and AUs can be *simultaneously* activated.

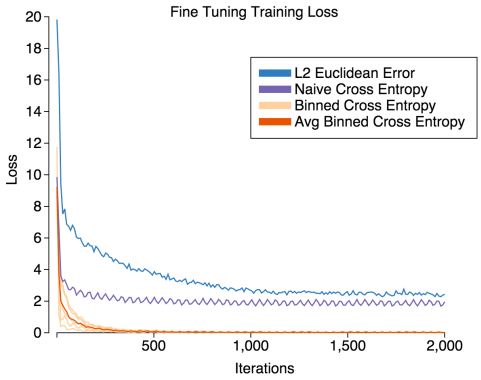


Fig. 4: The training loss on a single fold the CK+ dataset using three loss functions, L2 euclidean loss, naive cross entropy, and the binned cross entropy. The average binned cross entropy is the mean of the five binned AU training loss values over 2000 mini-batches of 200 samples.

Also, AU classifications can be real valued intensities, falling in the range from 0 to 5. Thus, the most logical approach to solving this problem would be to use an L2 euclidean regression loss, similar to [17].

Real Labeled Intensity L2 Regression - Using a regression loss, the network is able to estimate the real valued intensity of an AU. The L2 (euclidean) loss, L, is defined as,

$$L = \frac{1}{2N}\sum_{n=1}^{N} \|\hat{y_n} - y_n\|^2 \tag{1}$$

where n is a label in N training labels, $\hat{y} \in [-\infty, +\infty]$ are the predictions from the last layer and $y \in [0, 5]$ are the output intensity labels. Although logically and theoretically sound, minimizing L2 loss is much harder to optimize in practice than a more stable alternative. Intuitively, one can understand why this task is difficult as the L2 regression is attempting to output exactly one correct value; whereas in other losses, the magnitude of the predictions are what determines the output and the precise score is not as important. Another problem is that L2 losses are also less robust to outliers which may introduce very large gradients in training.

Multi-label, Naive Sigmoid Cross Entropy Loss - A second approach to the problem that has a more stable loss and can elegantly handle multiple simultaneous activations is the the sigmoid cross entropy loss. In the standard case of multi-label classification, y is the label indicating that the AU exists in the image and y is binary, $y \in \{0, 1\}$, then we define the cross entropy loss, L, as the following equation,

$$L = \frac{-1}{n}\sum_{n=1}^{N}[y_n \log \hat{y_n} + (1 - y_n)\log(1 - \hat{y_n})] \tag{2}$$

where $\hat{y}$ represents the sigmoid function, $\hat{y} = \frac{1}{1+e^w} \in [0, 1]$ and w represents the output of the last layer of our network, $w \in [-\infty, +\infty]$.

However, our application is not standard because we have real valued labels that represent the intensity of the AU present in the face image. Thus, a naive approach of using the sigmoid cross entropy loss is to use a new $y \in [0, 1]$, and map the intensity values to this range. An AU intensity of 5 would map to 1, 4 would map to 0.8, 3 would map to 0.6, etc. Although numerically, the gradient during back propagation will be computed correctly from a sigmoid calculation, the training loss will fail to converge due to the network's inability to accurately target non-binary labeling schemes, see Figure 4 for an illustration of the problems with both L2 and naive cross entropy losses.

Binned Cross Entropy Loss using Imputed Intensity - We propose a solution that overcomes both the instability of L2 regression and the non-binary labeling issue in the cross entropy loss by quantization e.g. BinCNN. For each intensity level, I, we threshold that intensity level and intensities greater than the threshold to a binary intensity mask, $\hat{I}$. Originally, we had only masked out a specific intensity; however, that turned out to be much too constrained. For this reason, we include any intensity above the threshold. Mathematically, this is simply represented as,

$$\hat{I} = \begin{cases} 1 & \text{if } I >= t \\ 0 & \text{if } I < t \end{cases} \tag{3}$$

where t is a threshold value from 1 to 5. For each intensity mask, we fine-tune a neural network with a standard binary sigmoid cross entropy loss. Intuitively, one can think of this binned process as building five classifiers, one for each thresholded intensity level. For our classification task, we combine the sigmoid output results of the five CNNs into a final output which is a maximum over the five outputs. For example, if the outputs of the CNNs are, $\rho = \{o1, o2, o3, o4, o5\}$, then the final output is $final = \max(\rho)$, where $final \in [0, 1]$. The training loss of the five binned CNNs and the average of these losses can be seen in Figure 4.

F. Semantic Context Model using FACS Co-Occurrence

The results of the binned cross entropy method are passed to our semantic context model (SBinCNN). Our model is able to refine the results from the neural network by utilizing a fully connected probabilistic conditional random field (CRF). The unary potentials of the CRF are defined by the neural network, and the pairwise potentials are computed by estimating the parameters that maximize the likelihood of the training data. The general form of the semantic context model for a given image, Im, is the following,

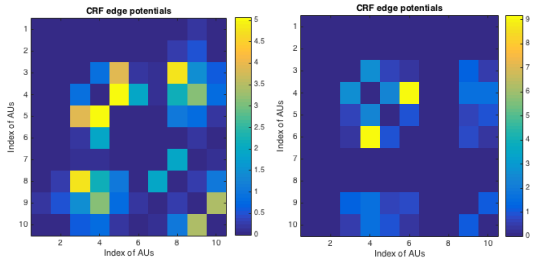
$$p(f_1 \dots f_n | Im) = \frac{1}{Z} E(f_1 \dots f_n) \prod_{i=1}^{|N|} p(f_i | Im) \tag{4}$$

where $f_{n \in N}$ are the AUs being classified, $p(f_i)$ are the unary potentials obtained from BinCNN, Z is the normalization (partition) constant, and $E(\cdot)$ is the edge interaction function defined by,

$$E(f_1 \dots f_n) = \exp(\sum_{i,j=1}^{|N|} w_e \phi_e(i, j)) \tag{5}$$

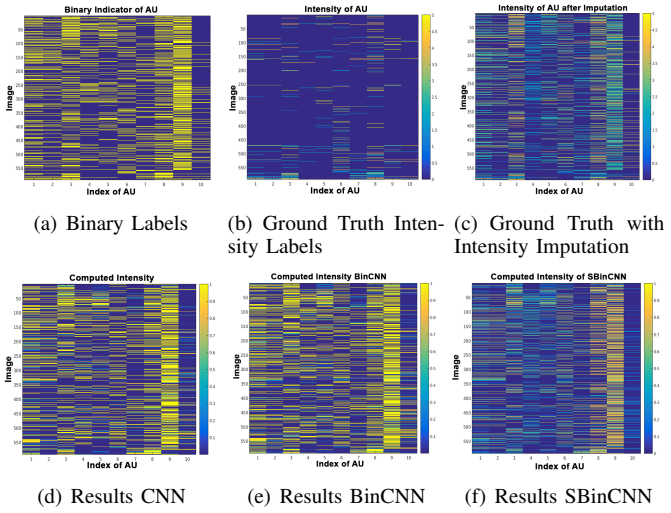
TABLE I: Description of the 10 Action Units (AU) we used from the Facial Action Coding System.

AU	Definition	AU	Definition
1	Inner brow raiser	12	Lip corner puller
2	Outer brow raiser	15	Lip corner depressor
4	Brow lowerer	17	Chin raiser
6	Cheek raiser	25	Lip apart
7	Lid tightener	26	Jaw drop



(a) CRF CK+ (b) CRF PainAU

Fig. 5: Visualization of the pairwise edge potentials of a training data partition on our datasets computed by maximum likelihood estimation.



(a) Binary Labels (b) Ground Truth Intensity Labels (c) Ground Truth with Intensity Imputation

(d) Results CNN (e) Results BinCNN (f) Results SBinCNN

Fig. 6: Label intensity matrices for the CK+ dataset and results of our methods. Our SBinCNN results (f) are more in line with the ground truth (b) and ground truth imputed intensity labeling (c) (view in color).

In this equation, w_e is a weight parameter for the specified edge, and $\phi_e(\cdot)$, are the edge interaction potentials computed from the training data. A value (i, j) in the edge potential matrix, see Figure 5, represents the strength of the relationship between AUs. The final partition function and marginal probability of each AU can be computed by exact inference.

IV. Experiments and Results

We perform two sets of experiments, intensity classification (strength of the AU), and binary classification (presence or absence of the AU). The 10 AUs we considered are the following, {1,2,4,6,7,12,15,17,25,26} and their descriptions can be seen in Table I.

A. Intensity Level Classification

Our first experiment we are classifying the intensity of the 10 AUs on our datasets. The intensity level has a range of 0-5, where 0 indicates that the AU is not present and 5 is the maximum activation of the AU. We compare our SBinCNN measure against a standard off-the-shelf CNN, as well as the BinCNN that does not incorporate our semantic context model. For the CK+ dataset, we can compute the L2 distance from the ground truth (non-imputed) AU intensity to our predicted results. The L2 distances are as follows: SBinCNN is 30.63, BinCNN is 39.72, and CNN is 40.75 (where lower is better). Similarly for the Pain AU database our L2 distances on a randomly sampled set of 1000 images from ten subjects are as follows: SBinCNN is 68.54, BinCNN is 77.13, and CNN is 78.21. Similar to our SBinCNN quantitative improvements, qualitatively, our method more closely matches the ground truth labels, see Figure 6.

B. Binary Classification Experiments and Evaluation methods

For the next set of experiments, we perform a binary classification for fair comparison against the state-of-the-art [13], [14], [12]. We compare our method, SBinCNN, with other recent works using the evaluation metrics, precision, recall, and the f1 measure defined as $f1 = 2 * \frac{precsion * recall}{precision + recall}$.

In Table II, the SVM-PHOG and SVM-CNN are custom matlab implementations of a single label AU SVM classification using the PHOG and CNN code features. 10 SVM models are created for the 10 different AUs. The reported results are obtained by a 10 fold cross validation on the CK+ dataset, and a leave-one-subject-out validation on the Pain AU dataset.

For the MCF [12] method, we use the published results of their algorithm obtained from [12] on this dataset. Because only the f1 scores are reported, we are do not have the precision and recall scores for this method. For BCS [13] and BGCS [14], we use the publicly available code for the methods and perform a similar 10 fold cross validation to obtain our final posted results. These methods use the PHOG image feature by default. We also experimented with simply replacing the PHOG feature in the BGCS framework with the CNN codes to see how this affects performance. We report the results of this experiment as the BGCS-CNN method.

Lastly, we present the results of our fine-tuned SBinCNN method which uses our proposed binned cross entropy loss and semantic context model. Our architecture is a fine-tuned seven layer neural network trained using stochastic gradient descent on a 10 fold cross validation partitioning scheme, see Table II and Figure 7.

V. Discussion and Conclusion

Our results show that our SBinCNN method is able to outperform all of the experimental methods and state-of-the-art in both datasets. And as predicted, the CNN codes are able to improve existing classification methods when compared to hand crafted features like PHOG. It is also apparent from our results that models which can take advantage of the co-occurrences of AUs have substantial benefits over single label classifiers.

TABLE II: Precision, recall, and f1 scores for two datasets, CK+ and the UNBC-McMaster (Pain AU) dataset. Our Semantic Binned Convolutional Neural Network (SBinCNN) performs better than CNN codes within an existing model (BGCS-CNN, SVM-CNN), a stand alone CNN [4], standard binned CNN (BinCNN), and other state-of-the-art methods [13], [12], [14].

Dataset	Metric	SVM-PHOG	SVM-CNN	BCS [13]	MCF [12]	BGCS [14]	BGCS-CNN	CNN [4]	BinCNN	SBinCNN
CK+	precision	0.70	0.73	0.67	-	0.67	0.75	0.78	0.76	**0.78**
CK+	recall	0.72	0.69	0.69	-	0.73	0.80	0.79	0.81	**0.82**
CK+	f1 score	0.71	0.71	0.67	0.76	0.69	0.77	0.78	0.79	**0.80**
Pain AU	precision	0.34	**0.41**	0.19	-	0.24	0.32	0.36	0.36	0.37
Pain AU	recall	0.39	0.38	0.36	-	0.40	0.55	0.53	0.61	**0.67**
Pain AU	f1 score	0.36	0.39	0.24	0.38	0.30	0.40	0.43	0.44	**0.48**

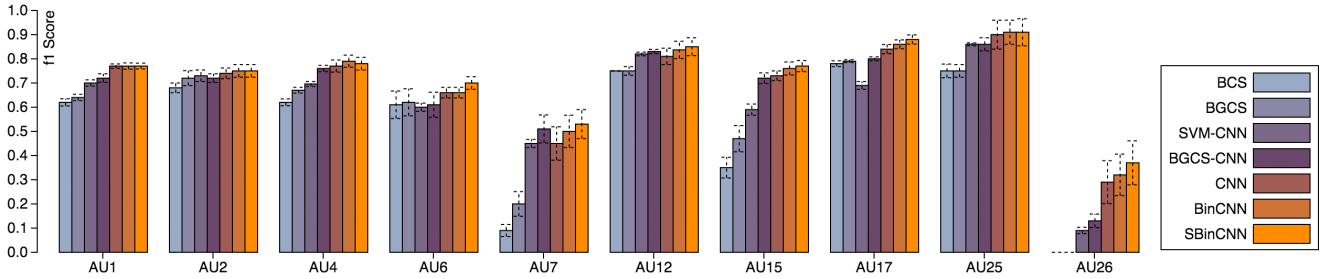


Fig. 7: f1 measurements of the 10 AUs measured from the CK+ dataset. The graph displays the mean and standard deviation of the AU measurements. Most of the f1 score improvements can be seen in low scoring AUs (6, 7, 26) where the semantic model is increasing the likelihood of the AU due to co-occurrences.

In the one case where the SVM-CNN appears to be out performing other methods in the Pain AU dataset (see Table II), we would like to highlight that the overall f1 score, or harmonic mean between precision and recall, is the measure that was maximized. One can artificially inflate the precision by sacrificing recall, and vice versa. Thus, although the precision in the SVM-CNN is the highest among our evaluated algorithms, the f1 score of the SBinCNN is approximately 23% higher than the maximum f1 SVM-CNN score.

In conclusion, we demonstrate the effectiveness of solving a real valued multi-AU intensity L2 regression problem with a novel binned loss and semantic model. We can handle missing intensity data through a regression based imputation. Further, we demonstrated the quantitative and qualitative benefits of utilizing deep learning and semantic neural networks (SBinCNN) towards Action Unit classification, and further, show notable improvement over the current state-of-the-art.

Acknowledgment

Thank you to Dr. Jesse Frey for his insight on the semantic context model. This work was supported by an AWS in Education Grant award.

References

[1] A. Mehrabian, *Nonverbal communication*, Transaction Publishers, 1977.

[2] P. Ekman and EL Rosenberg, *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*, Oxford University Press, 1997.

[3] A. Krizhevsky, I. Sutskever, and G.E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[4] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*, 2014, pp. 675–678.

[5] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[6] T. Kanade, J. F Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *IEEE Conference on Automatic Face and Gesture Recognition*, 2000, pp. 46–53.

[7] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in *Computer Vision and Pattern Recognition Workshops*, 2010, pp. 94–101.

[8] K. M Prkachin and P. Solomon, "The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain," *Pain*, vol. 139, no. 2, pp. 267–274, 2008.

[9] P. Lucey, J. Cohn, K. Prkachin, P. Solomon, and I. Matthews, "Painful data: The unbc-mcmaster shoulder pain expression archive database," in *IEEE International Conference on Automatic Face & Gesture Recognition and Workshops*, 2011, pp. 57–64.

[10] Z. Zeng, M. Pantic, G. Roisman, T.S Huang, et al., "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 39–58, 2009.

[11] A. Mahalanobis, BVK Vijaya K., and D. Casasent, "Minimum average correlation energy filters," *Applied Optics*, vol. 26, no. 17, pp. 3633–3640, 1987.

[12] S. W Chew, S. Lucey, P. Lucey, S. Sridharan, and J. Conn, "Improved facial expression recognition via uni-hyperplane classification," in *Computer Vision and Pattern Recognition*, 2012, pp. 2554–2561.

[13] A. Kapoor, R. Viswanathan, and P. Jain, "Multilabel classification using bayesian compressed sensing," in *Advances in Neural Information Processing Systems*, 2012, pp. 2645–2653.

[14] Y. Song, D. McDuff, D. Vasisht, and A. Kapoor, "Exploiting sparsity and co-occurrence structure for action unit recognition," *International Conference on Automatic Face and Gesture Recognition*, 2015.

[15] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie, "Objects in context," in *IEEE International Conference on Computer vision*, 2007, pp. 1–8.

[16] Y. Tian, T. Kanade, and J. Cohn, "Recognizing action units for facial expression analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 97–115, 2001.

[17] A. Gudi, H E Tasli, T. M den Uyl, and A. Maroulis, "Deep learning based facs action unit occurrence and intensity estimation," *International Conference on Automatic Face and Gesture Recognition*, 2015.

[18] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *International Conference on Image and Video Retrieval*, 2007, pp. 401–408.

[19] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in Neural Information Processing Systems*, 2014, pp. 3320–3328.