Otto-von-Guericke University Magdeburg

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

Neuro-Information Technology Department
Institute for Information Technology and Communications (IIKT)

Research Track Work

Facial Expression Recognition Under Partial Occlusion

Author:
ESAM M. SHARAF

September 30, 2019

Advisers:

Supervisor
Prof. Dr.-Ing. habil. Ayoub Al-Hamadi

Neuro-Information Technology Department
Otto-von-Guericke University
Universitätsplatz 2
39106 Magdeburg, Germany

Supervisor
Dipl.-Ing.-Inf. Philipp Werner

Neuro-Information Technology Department
Otto-von-Guericke University
Universitätsplatz 2
39106 Magdeburg, Germany

M. SHARAF, ESAM:
Facial Expression Recognition Under Partial Occlusion
Research Track Work, Otto-von-Guericke University
Magdeburg, 2019.

Contents

Abstract

Action Unit (AU) taxonomy has played a great role in facial expression recognition and human computer interaction algorithms by the robust definitions of biology and muscles motion. This work investigates using Deep Convolutional Neural Network (DCNN) to utilize AU-based facial expression recognition under occlusion. The occurrence of AU12 was examined through two separate DCNN pre-trained models by using Keras framework. Also, the intensity estimation of AU12 and AU4 was tested based on the same pre-trained models, on the case of non-occlusion as well as occlusion existence, by non-linear regression classifier. The Bosphorus 2D images database used for training the models and validating recognition performance. F1 score and Intraclass Correlation Coefficient were adopted for evaluation. This work's results, with the simplicity of pre-processing stage, indicate good level of recognition for AU12. On the other hand, AU4 recognition is more challenging and more research is required in the future.

1
Introduction and Motivation

1.1 Motivation

Facial expression can give valuable insights into human's feelings and health. In the medical domain, it may be used to automatically monitor the pain, vigilance, emotions as well as supporting diagnosis of depression. In addition, it may be used for enhancing contact-free human-machine interaction; during surgery for example.

Partial face occlusions occur often in medical context; due to respiratory support (masks and tubes), bandages, surgical masks, and operating microscopes. All of these counteract effective real-time tracking of patient/operator state in any relevant process, including face appearance and facial signals as being valued information source that can be availed in many non-intrusive applications.

1.2 Aim of this Work

This work aim is to analyze DCNN for pain expression recognition from the Bosphorus database under partial occlusions, in the terms of Action Unit method of perception of facial expressions. Two AUs were selected for this experiment which are Lip Corner Puller (AU12) and Brow Lowerer (AU4), in the purpose of examining how good two pre-trained DCNN models offered by Keras framework. Their corresponding weights were being fine-tuned instead of being freezed along the training phase. In addition, the work investigates any significant link between the selected AUs under variant occluder conditions and introduces points for consideration at the end of this work.

1.3 Structure of the Content

The remained sections were structured as follows: Chapter 2 covers related works which the experiment relies upon their conclusions. Chapter 3 includes details of the proposed DCNN models. Chapter 4 provides results of each contribution of this work. Chapter 5 concludes the results and highlights points for future work.

2
Related Work

Action Unit (AU)-based methods have been used for optimizing different models that are able to describe and analyze facial expression building on the solid definition of AUs. AUs represent a facial expression as a result of underlying muscles motions based on Facial Action Coding System (FACS) [10]. In recent years, Convolutional Neural Networks have (CNN) shown strong impact on field of face expression recognition with action units, as blueprint to decode number of expressions like those associated with happiness and sadness.

In [15], the authors showed the ability of zero-bias CNN to learn features corresponded to AUs by visualizing the spatial patterns that maximally excited different filters in CNN network designed for that. The work achieved 89.8%(±1.8%), 98.3%(±1.6%) recognition accuracy on Toronto Face Dataset (TFD) [27], extended Cohn-Kanade(CK+) Dataset [18] respectively.

In [30], an ensemble of Deep Convolutional Neural Networks (DCNNs), for image base static FER, were used for automatically classify 7 emotions. The ensemble network basically contains five convolutional layers, three stochastic pooling layers and three fully connected layers. The network pre-trained on FER-2013 Challenge [2] and fine tuned on Static Facial Expressions in the Wild (SFEW) dataset [8]. The proposed frameworks, likelihood loss and hinge loss, achieved 55.96% and 61.29% accuracy rate respectively on SFEW dataset showing the considerable potential of CNNs in FER.

In addition, [4] exhibits that CNN can be competitive at very high automatic expression recognition rates. The proposed CNN architecture surpasses earlier high results with 99.6%, 98.63% for Extended Cohn-Kanade

(CKP) and MMI datasets [22] respectively. [11] investigated the ability of novel convolutional neural network to recognize facial AUs occurrences on two datasets as well as their intensity estimation in form of binary and continuous output orderly. The study pointed to more investigation required on CCNs structure on large datasets to obtain higher recognition rates.

FER under occlusion is still challenging task comparing to natural human ability to extract various expressions from faces in uncontrolled environments. Recently, Region Attention Network (RAN) method was presented by [29] with state-of-art results. It basically captures the importance of facial regions for occlusion, then aggregating varied number of regional CNN-based features into certain representation. Along with facial expressions are defined by several AUs, Region Biased (RB-Loss) function was proposed. The method achieved accuracy rates 89.16% (RAN-VGG16[23]), 59.5% (RAN-ResNet18 [13] with oversampling), 86.9% (RAN-ResNet18), and 56.4% (RAN(VGG16+ResNet18)), on FERPlus [3], AffectNet [19], RAF-DB [17], and SFEW [9] datasets respectively. Considering RAN network with RB-Loss only, the accuracy measure of the case of occlusion, compared to a baseline was enhanced from 73.33% to 83.63% on FERPlus, 49.48% to 58.50% on AffectNet, 80.19% to 82.72% on RAF-DB.

Another proposed CNN with attention mechanism (ACNN) showed state-of-art results on both real and synthetic occlusions. Two versions of ACNN were introduced; global local based (gACNN) and patch based (pACNN). Among various evaluations, where the proposed networks tested on three In-The-Lab datasets, gACNN model that was trained on AffectNet dataset achieved 91.64/88.17(%), 70.37/65.48(%), 58.18/55.42(%) accuracy rate on CK+, MMI [21] and Oulu-CASIA [28] respectively with original/occluded images trials.

3
Methods

As being an experimental research, this work examined implementing Keras [5] with TensorFLow [1] backend through transfer learning and fine tuning for two pre-trained CNN structures, which are ResNet50 [12] and MobileNetV2 [14]. Both models own pre-trained weights on ImageNet [7]. A classifier of two layers extended the network, one for flattening the collected features, and one for the output prediction. Two metrics, F1 score and Intraclass Correlation Coefficient (ICC(3,1)) [25], were selected for performance evaluation. Also, two kinds of occlusions pasted over face's parts simulating common facial occluders in the medical field. The Bosphorus database [24] was selected for all contributions of this work.

The main contributions of this work:

1. Using ResNet50 pre-trained model as a binary classifier for detecting the occurrence of AU12.

2. Using MobilNetV2 pre-trained model as a binary classifier for recognizing the occurrence of AU12.

3. Non-linear regression implementation to predict AU12 and AU4 values individually in range from 0 to 5 through implementing MobileNetV2 pre-trained model.

4. Estimating the intensity value of AU12 and AU4 as being under occlusion. AU12 was investigated under three occluders; fully, half and small mask-like occluder. While AU4 was investigated under two different size glass-like occluders; big round occluder covers the eye and eye surrounding area including the eyebrows, and a small one covers only the eye.

3.1 Pre-processing

Since each model has predefined input size, The Bosphorus dataset images' dimensions reduced accordingly. ResNet50 and MobileNetV2 both have default input size of 224x224x3. Since small image size expedites learning performance, input square float data type images of 150x150x3 for ResNet50 Model, 160x160x3 for MobileNetV2 was the input image size in training and validation phases.

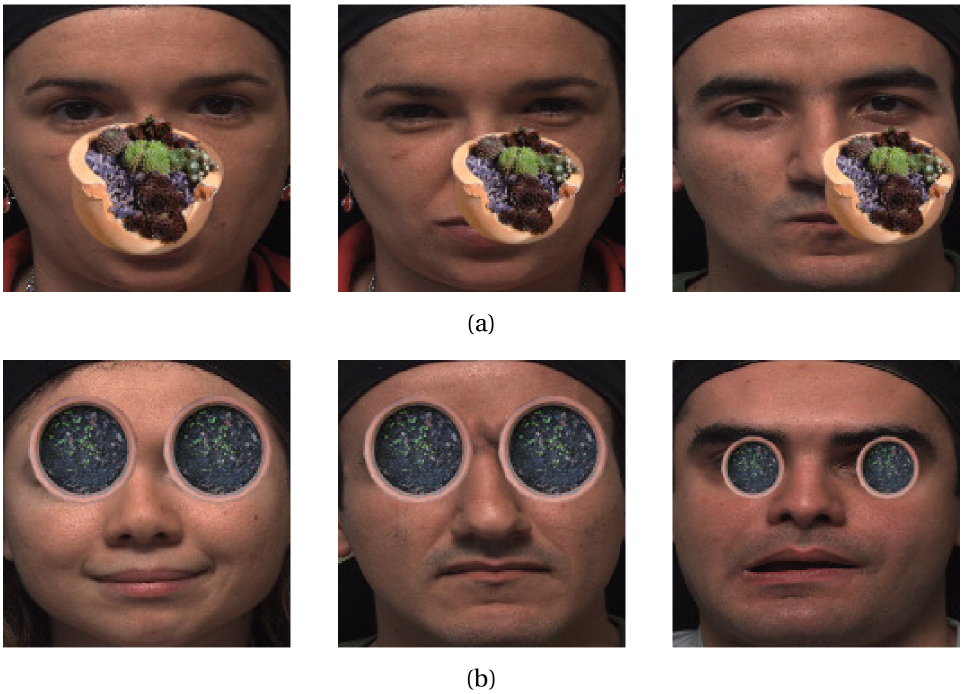
Pre-processing stage additionally included data augmentation like horizontal and vertical flipping as well as range of brightness adjustments (0.5-1.5) to encourage the model to generalize.

Additionally, pasting two kinds of occluders on the images to hide action units related to pain recognition (AU12 & AU4) was a further pre-processing step added for the regression model (section 3.4). Fig. 3.1 shows images from the Bosphorus dataset under partial occlusion, which mimic what can be found in the medical field, particularly face inhalator and glasses.

3.2 ResNet50 Model As Binary Classifier

Residual Networks (ResNets) consider as an example of very deep classic structure in computer vision literature. ResNet50, by its name, has fifty layers. The basic component called "Convolutional Block" which ensembles a 2D convolutional layer, Batch Normalization as well as a Rectified Linear Unit (ReLU) [20] layers. Another essential entity of Residual Networks is "Identity Block" which forwards layer's output by skip connection technique. The way ResNets were designed is to solve the "vanishing gradient" problem in DCNN networks. ResNet won the ImageNet (ILVRC2015) challenge. By that, it was initially selected for investigating AU12 occurrence based on 2D facial images from the Bosphorus database.

The high number of layers normally leads to early overfitting and makes the model impractical. Large capacity networks, like ResNet50, are usually learning fast on the training data given, but not with an approximate rate on the validation data. In AUs pattern recognition, when a network in an overfitting state, it learns irrelevant features that act like



(a)

(b)

Figure 3.1: Pre-processing examples: (a) Inhalator-like occlusion at different positions. (b) Glass-like occlusion of a big size (left and middle) hiding AU4 and a small size keeping the eyebrows manifested (right).

a noise and degrade AU recognition accuracy on the unseen data. Experimentally, two trials were performed with different cut-off layers, The first one has a branch-out to the top binary classifier after "activation_5" layer, while the other has the branch-out after "activation_6" layer. That with other features were the same in the two trials, F1-score decreased significantly from 0.72 to 0.51 by making the cut-off at "activation_6".

Fig. A.1 shows ResNet model structure as the cut-off was made at "actination_5" layer. This structure was used for AU12 detection. The weights, up to cut-off layer, were fine-tuned in order to urge the network to generalize.

The corresponding classifier on the top of the network consists of flatten layer, and a single-neuron output layer with Sigmoid activation function [6], without any regularization techniques like Dropout [26]. In addition, binary_crossentropy function was selected as a loss function. This structure, shown in Fig. A.1, was concluded after extensive experiments.

3.3 MobileNetV2 Model As Binary Classifier

MobileNetV2 represents the opposite example of ResNet50; it has a light weight architecture and so a faster performance, which make it suitable for limited-memory mobile application. The basic idea of MobileNetV2 is to replace a full convolutional operator with a factorized version that splits the convolution into two separate layers, the first layer is called a depthwise convolution; which performs lightweight filtering by applying a single convolutional filter per input channel. The second layer is a 1×1 convolution, called a pointwise convolution [14].

Fig. A.2 depicts the detailed structure of MobileNetV2 model of which the highest F1-score achieved for binary classification, to classify whether an image has AU12 pattern or not. As seen in Fig. A.2, a cut-off made after block_3_depthwise_relu layer for branching out to the classifier. The classifier consists of a flatten layer, Dropout layer of value equals to 0.4 followed by an output layer with Sigmoid activation function. This model is the result of several features modifications upon MobileNetV2 structure for acquiring the best results.

3.4 MobileNetV2 Model For AU Intensity Estimation without occlusion

This Model has the same basic MobileNetV2 structure of previous section but with a linear activation function at the output layer. The intensity of AU12 was predicted as continuous value from 0 to 5 according to the Bosphorus dataset AU intensity levels. Mean Square Error (MSE) was chosen as a loss function. ICC(3,1) metric was picked for comparing the predicted values and ground truth labels of the target AU.

3.5 MobileNetV2 Model For AU Intensity Estimation under partial occlusion

As it is likely in medical environment to fix devices, masks or glasses on users' faces, these in turn occlude one or more AUs that are involved in pain expression assessment. By that, extracting pain expression from non-occluded regions becomes essential. In this part, Lip Corner Puller (AU12) and Brow Lowerer (AU4) were occluded by two occluders through a pre-processing function. For this part, the exact model that was used for non-occlusion trials, was executed here for fair evaluations.

AU12 was occluded by inhalator-like occluder. By changing the parameters in the pre-processing function, the facial images occluded at three different positions on the image grid to find out any significant difference among their output results, and with that of non-occluded images as well. Fig. 3.1 (a) shows the output AU12-occluded images by the pre-processing stage. Predicting AU12 values while they are under full occlusion gives an insight how much good/bad other face regions contribute in AU12 intensity estimation.

Frowning is another sign of pain expression. AU4 was occluded by glass-like occluder of two sizes; a small one covers only the eye hole and a big-sized one occludes all the areas around the eye including AU4 and the eyebrows. The small-sized occluder was applied to mimic real examples like small personal glasses, also to examine the adverse effect -if there is- of the eye hole occluder on AU4 intensity estimation.

The research was extended by predicting AU12 intensity labels while AU4 was under occlusion, to conclude any significant role AU4 has on AU12 intensity estimation. Also in a reverse way, AU4 values were estimated while AU12 was occluded.

3.6 F1- Score and Intraclass Correlation Coefficient (ICC3,1)

For more reliable evaluation, F1-score and ICC(3,1) were selected. Both are not provided by Keras framework. F1 measure was used for evaluating the binary classifier. It is calculated as follows:

$$F_1 = \frac{2PR}{P+R} \cdot \tag{3.1}$$

where P is the precision and R is the recall.

For AUs intensity estimation, $ICC(3,1)$ metric was adopted based on the formula mentioned in [25]. With number of raters K equals 2, Between Targets Mean Squares BMS, Residual Mean Squares EMS, it is calculated:

$$ICC(3,1) = \frac{BMS - EMS}{BMS + (K-1) * EMS} \cdot \tag{3.2}$$

In all trials, both F1 score and ICC(3,1) were computed per epoch not per batch.

4
Experiments and Evaluation

In this section, I provide the best results achieved on the Bosphorus database. 2D images were fed in for training and validating each one of the four models mentioned in methods chapter. The Bosphorus dataset comprises 105 individuals. The images are of realistic multi-expression mutlipose faces, some with partial occlusions. Only Portion of the Bosphorus images have been FACS-coded. For this work, the set of input images comprise facial images of upper action units, lower action units and a combination of both. Only images that exhibit actions units and emotions were listed for training and validation phases. The list of images was divided into 2011 images for training phase and 893 for validation phase. No mutual images of the same individual between the training and validation image subsets.

4.1 Binary Classification Evaluation

Table 4.1 shows F1 score for RestNet50 and MobilNetV2 models for detecting the presence of AU12. Two optimizers have chosen for learning; Adam and Stochastic Gradient Descent(SGD) with different learning rates. Also, data augmentation techniques was applied in the pre-processing to prevent overfitting. The two proposed models for binary classification have good results although the ground truth labels of AU12 suffers from high skewness to zero class (80% of total binary labels). As shown in the table, no notable difference in F1 score results between ResNet50 and MobileNetV2 pre-trained models. As well, the classification metrics of Adam optimizer are close to that of Stochastic Gradient Descent (SGD), regardless the implemented model. Vertical augmentation has no negative effect in identifying AU12 occurrence contrary to what had been expected.

Additionally, a small database was created for testing MobileNetV2 binary model. It contains 34 images of neutral and expressive faces collected randomly, part of them under occlusion, Fig 4.1 shows samples of the images. Fig. 4.2 is the confusion matrix of the resulted binary classifications by MobileNetV2 model. The results point to more improvements on the proposed DCNN models is necessary, since the model failed to identify AU12 occurrence for the majority of images (0.65%). The model suffers from low recall metric over the test images, specifically images of class 1.

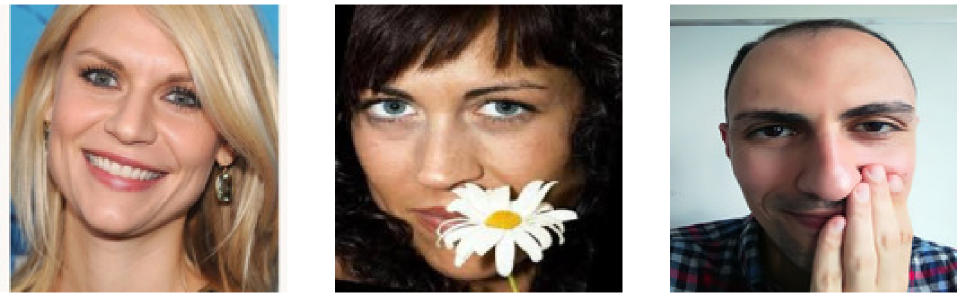


Figure 4.1: Examples of images for testing the MobileNetV2 binary classifier.

Table 4.1: F1 score values comparison between the two models used for binary classifications for AU12 occurrence. Third row describes data augmentation techniques as a pre-processing (H = Horizontal flip, V = Vertical flip, B = Brightness adjustment).

Model	Optimizer	Augmentation	F1-score
ResNet50	Adam(0.0001)	H/V/B	0.72
ResNet50	SGD(0.0001)	H/V/B	0.70
MobileNetV2	Adam(0.0001)	H/B	0.68
MobileNetV2	SGD(0.001)	H/V/B	0.72

4.2 Intensity Estimation Evaluation

Intensity estimation of AU12 and AU4 values, in range from 0 to 5, were predicted by a regression model. Each action unit of them was investigated with several different occlusions. Since both AU12 and AU4 related

		Actual value		
		0	**1**	**total**
Prediction outcome	**0**	11	21	32
	1	1	1	2
	total	12	22	

Figure 4.2: Confusion matrix of 34 images as test data for the proposed MobileNetV2 binary classifier.

to pain assessment, the investigation was extended for any significant link about predicting one action unit as the other one being under occlusion.

Table 4.2 shows the correlation measure of Lip Corner Puller AU (AU12) with different positions of mask-like occlusion as stated in the table as full, half, small occlusion. Each state is shown in Fig. 3.1 (a) left, middle, right respectively. Logically, the no-occlusion case has the highest correlation with 0.78. According to [16] guideline, this result indicate good correlation between AU12 true labels and the predicted labels of it.

The rest of results in table 4.2 portray the negative effect of the occlusion on AU12 intensity estimation at three levels. Fully occluded case has the highest reduction on F1 measure (0.25) compared to the non-occlusion case. Similarly, half and small occlusion cases also have decreased the recognition rate, 0.23 and 0.19 respectively. Obviously, the occlusion existence of it self has significant negative effect on AU12 intensity estimation. This has been concluded by considering that unoccluded training/occluded validation images (T0/V1) case is the hardest case the algorithm would encounter, no matter how large the occlusion as shown in the table. This conclusion holds true, except the half occlusion case where T0/V1 has slightly better F1 score than T1/V0.

More over, by looking at T1/V0 cases through table 4.2 where higher rates achieved compared with T0/V1 cases, indicates that the trained model was able to learn from the unoccluded regions in a significant sense. The same thing for T1/V1 trials with significant enhancement when they are compared to T0/V1 combinations no matter the state of occlusion. In addition, the same table shows that the smaller the occlusion, the more MobileNetV2 model is successful in AU12 intensity estimation at any T/V combination.

Table 4.3 includes results of estimating AU12 values while AU4 occluded by two glass-like occluders, which are different in size. In the full occlusion case, the correlations were almost the same with no significant reduction across the corresponding T/V combinations. On the other hand, a small reduction in the metric was noticed with small eye hole occlusion case as shown int the table, especially T1/V1 case. This reduction may be caused by the inaccurate process of locating the occluder on the target region (i.e. the eye hole) at the pre-processing. As long as the pre-processing function pastes the occluder at fixed pixels values while the input images have slightly different poses, this could have led to put the occluder slightly up or off the eye hole. As result to that, the occlusion manipulated the facial pattern and so the intensity estimation.

Moreover, ICC(3,1) metric values are listed in table 4.4 for Brow Lowerer Action Unit (AU4) intensity estimation. On contrary to AU12 recognition results, the correlation was generally poor and the model failed in distinguishing the targeted AU4 in all cases including the non-occlusion case. In spite of that, the results indicates that the model depends on non-occluded regions to estimate AU4 value. This appears on T0/V1 in the case of full occlusion which has a bit higher recognition rate compared with the non-occlusion one. Following that, all the cases where the training images were under occlusion have had significant higher correlation rate, only T1/V0 case at fully occlusion state has a metric reduction about 0.04 compared with the non-occlusion case. All in all, no clear statement can be set up by the obtained poor correlation results about how good/bad the implemented occlusions influenced AU4 intensity estimation.

One more trial was for predicting AU4 while AU12 under occlusion, It turned out that occlusion existence, in all cases, significantly enhanced model performance on AU4 intensity estimation compared with the non-occlusion case, in variant levels as shown in table 4.5. In other words, the region, which was covered by the occluder, disturbs MobileNetV2 network output to the right AU4 value.

Table 4.2: ICC(3,1) correlation comparison of intensity estimation for Lip Corner Puller Action Unit (AU12) under different facial occlusion levels (fully /half /small) and no-occlusion case. T/V: states the case of training/ validation sets being under occlusion (1) or not (0). Features: H: Horizontal flip, B: Brightness adjustment, Max: Max pooling layer, Flat: Normal flat layer.

Lip Corner Puller (AU12)	T/V	Optimizer	Features	ICC(3,1) (AU12)
No occlusion	T0/V0	Adam(0.001)	H/B/Max	0.78
Full occluded	T0/V1	Adam(0.001)	H/B/Flat	0.53
	T1/V0	SGD(0.0001)	H/B/Max	0.61
	T1/V1	Adam(0.001)	H/B/Max	0.56
Half occluded	T0/V1	Adam(0.001)	H/B/Max	0.55
	T1/V0	Adam(0.001)	H/B/Flat	0.53
	T1/V1	Adam(0.001)	H/B/Flat	0.63
Small Occlusion	T0/V1	Adam(0.001)	H/B/Flat	0.59
	T1/V0	Adam(0.001)	H/B/Flat	0.68
	T1/V1	Adam(0.001)	H/B/Flat	0.66

Table 4.3: ICC(3,1) correlation comparison of intensity estimation for Lip Corner Puller Action Unit (AU12) with glass-like occluder hides AU4 area (fully, eye hole) and the initial no-occlusion case. T/V: states the case of training/ validation sets being under occlusion (1) or not (0). Features: H: Horizontal flip, B: Brightness adjustment, Max: Max pooling layer, Flat: Normal flat layer.

Brow lowerer (AU4)	T/V	Optimizer	Features	ICC(3,1) (AU12)
No occlusion	T0/V0	Adam(0.001)	H/B/Max	0.78
Fully Occluded	T0/V1	Adam(0.001)	H/B/Flat	0.79
	T1/V0	Adam(0.001)	H/B/Flat	0.75
	T1/V1	Adam(0.001)	H/B/Flat	0.77
Eye hole Occluded	T0/V1	Adam(0.001)	H/B/Flat	0.74
	T1/V0	Adam(0.001)	H/B/Flat	0.73
	T1/V1	Adam(0.001)	H/B/Flat	0.69

Table 4.4: ICC(3,1) correlations of intensity estimation for Brow Lowerer Action Unit (AU4) at no-occlusion, fully and eye hole occlusion cases. T/V: states the case of training/ validation sets being under occlusion (1) or not (0). Features: H: Horizontal flip, B: Brightness adjustment, Max: Max pooling layer, Flat: Normal flat layer.

Brow Lowerer (AU4)	T/V	Optimizer	Features	ICC(3,1) (AU4)
No occlusion	T0/V0	Adam(0.001)	H/B/Max	0.30
Fully occluded	T0/V1	Adam(0.001)	H/B/Flat	0.32
	T1/V0	Adam(0.001)	H/B/Flat	0.26
	T1/V1	Adam(0.001)	H/B/Flat	0.36
Eye hole occluded	T0/V1	Adam(0.001)	H/B/Flat	0.24
	T1/V0	Adam(0.001)	H/B/Flat	0.43
	T1/V1	Adam(0.001)	H/B/Flat	0.45

Table 4.5: ICC(3,1) correlation comparison of intensity estimation for Brow Lowerer Action Unit (AU4) with mask-like occluder hides AU12 at three level of occlusion (fully /half /small) and the initial no-occlusion case. T/V: states the case of training/ validation sets being under occlusion (1) or not (0). Features: H: Horizontal flip, B: Brightness adjustment, Max: Max pooling layer, Flat: Normal flat layer.

Lip Corner Puller (AU12)	T/V	Optimizer	Features	ICC(3,1) (AU4)
No occlusion	T0/V0	Adam(0.001)	H/B/Flat	0.30
Fully occluded	T0/V1	Adam(0.001)	H/B/Flat	0.49
	T1/V0	Adam(0.001)	H/B/Flat	0.41
	T1/V1	Adam(0.001)	H/B/Flat	0.46
Half occluded	T0/V1	Adam(0.001)	H/B/Flat	0.48
	T1/V0	Adam(0.001)	H/B/Flat	0.54
	T1/V1	Adam(0.001)	H/B/Flat	0.45
Small occlusion	T0/V1	Adam(0.001)	H/B/Max	0.51
	T1/V0	Adam(0.001)	H/B/Max	0.50
	T1/V1	Adam(0.001)	H/B/Max	0.51

5 Conclusions and Future Work

5.1 Conclusions

In this work, two DCNN pre-trained models were selected in order to recognize two AUs with 2D input images by using Keras framework. Custom classifier was built on the top of the models for extracting features for detecting AU12 occurrence, and estimating the intensity of AU12 and AU4. The proposed method achieved good results for identifying AU12 occurrence on the Bosphorus dataset, but it was majorly unsuccessful on the small dataset designed for this work. Also, fair-good results were achieved by MobileNetV2 model for predicting AU12 values. Introducing partial facial occlusions significantly lowered AU12 intensity estimations, which was concluded through variant facial occlusions on AU12 region. Moreover, the experiments showed that no significant effect from applying the glass-like occluder on AU12 intensity estimation.

On the other hand, poor correlations were achieved for AU4 intensity estimation by the proposed MobileNetV2 model. Unexpectedly, applying glass-like occluder leveled up AU4 intensity estimation except one trial. Additional experiments illustrated the positive effect of inhalator-like occluder on AU4 intensity estimation. Further investigation in the future is needed for making clear conclusions about AU4 intensity estimation.

5.2 Future Work

In the future, we will try to benefit from 3D features for classifying 2D images as well as adding more to the pre-processing stage in favor of getting higher correlations.

A
Detailed Results

```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 150, 150, 3)  0
__________________________________________________________________________________________________
conv1_pad (ZeroPadding2D)       (None, 156, 156, 3)  0           input_1[0][0]
__________________________________________________________________________________________________
conv1 (Conv2D)                  (None, 75, 75, 64)   9472        conv1_pad[0][0]
__________________________________________________________________________________________________
bn_conv1 (BatchNormalization)   (None, 75, 75, 64)   256         conv1[0][0]
__________________________________________________________________________________________________
activation_1 (Activation)       (None, 75, 75, 64)   0           bn_conv1[0][0]
__________________________________________________________________________________________________
pool1_pad (ZeroPadding2D)       (None, 77, 77, 64)   0           activation_1[0][0]
__________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)  (None, 38, 38, 64)   0           pool1_pad[0][0]
__________________________________________________________________________________________________
res2a_branch2a (Conv2D)         (None, 38, 38, 64)   4160        max_pooling2d_1[0][0]
__________________________________________________________________________________________________
bn2a_branch2a (BatchNormalizati (None, 38, 38, 64)   256         res2a_branch2a[0][0]
__________________________________________________________________________________________________
activation_2 (Activation)       (None, 38, 38, 64)   0           bn2a_branch2a[0][0]
__________________________________________________________________________________________________
res2a_branch2b (Conv2D)         (None, 38, 38, 64)   36928       activation_2[0][0]
__________________________________________________________________________________________________
bn2a_branch2b (BatchNormalizati (None, 38, 38, 64)   256         res2a_branch2b[0][0]
__________________________________________________________________________________________________
activation_3 (Activation)       (None, 38, 38, 64)   0           bn2a_branch2b[0][0]
__________________________________________________________________________________________________
res2a_branch2c (Conv2D)         (None, 38, 38, 256)  16640       activation_3[0][0]
__________________________________________________________________________________________________
res2a_branch1 (Conv2D)          (None, 38, 38, 256)  16640       max_pooling2d_1[0][0]
__________________________________________________________________________________________________
bn2a_branch2c (BatchNormalizati (None, 38, 38, 256)  1024        res2a_branch2c[0][0]
__________________________________________________________________________________________________
bn2a_branch1 (BatchNormalizatio (None, 38, 38, 256)  1024        res2a_branch1[0][0]
__________________________________________________________________________________________________
add_1 (Add)                     (None, 38, 38, 256)  0           bn2a_branch2c[0][0]
                                                                 bn2a_branch1[0][0]
__________________________________________________________________________________________________
activation_4 (Activation)       (None, 38, 38, 256)  0           add_1[0][0]
__________________________________________________________________________________________________
res2b_branch2a (Conv2D)         (None, 38, 38, 64)   16448       activation_4[0][0]
__________________________________________________________________________________________________
bn2b_branch2a (BatchNormalizati (None, 38, 38, 64)   256         res2b_branch2a[0][0]
__________________________________________________________________________________________________
activation_5 (Activation)       (None, 38, 38, 64)   0           bn2b_branch2a[0][0]
__________________________________________________________________________________________________
flatten_1 (Flatten)             (None, 92416)        0           activation_5[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 1)            92417       flatten_1[0][0]
==================================================================================================
Total params: 195,777
Trainable params: 194,241
Non-trainable params: 1,536
__________________________________________________________________________________________________
```

Figure A.1: ResNet50 pre-trained model utilized identifying AU12 occurrence.

```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 160, 160, 3)  0
__________________________________________________________________________________________________
Conv1_pad (ZeroPadding2D)       (None, 161, 161, 3)  0           input_1[0][0]
__________________________________________________________________________________________________
Conv1 (Conv2D)                  (None, 80, 80, 32)   864         Conv1_pad[0][0]
__________________________________________________________________________________________________
bn_Conv1 (BatchNormalization)   (None, 80, 80, 32)   128         Conv1[0][0]
__________________________________________________________________________________________________
Conv1_relu (ReLU)               (None, 80, 80, 32)   0           bn_Conv1[0][0]
__________________________________________________________________________________________________
expanded_conv_depthwise (Depthw (None, 80, 80, 32)   288         Conv1_relu[0][0]
__________________________________________________________________________________________________
expanded_conv_depthwise_BN (Bat (None, 80, 80, 32)   128         expanded_conv_depthwise[0][0]
__________________________________________________________________________________________________
expanded_conv_depthwise_relu (R (None, 80, 80, 32)   0           expanded_conv_depthwise_BN[0][0]
__________________________________________________________________________________________________
expanded_conv_project (Conv2D)  (None, 80, 80, 16)   512         expanded_conv_depthwise_relu[0][0
__________________________________________________________________________________________________
expanded_conv_project_BN (Batch (None, 80, 80, 16)   64          expanded_conv_project[0][0]
__________________________________________________________________________________________________
block_1_expand (Conv2D)         (None, 80, 80, 96)   1536        expanded_conv_project_BN[0][0]
__________________________________________________________________________________________________
block_1_expand_BN (BatchNormali (None, 80, 80, 96)   384         block_1_expand[0][0]
__________________________________________________________________________________________________
block_1_expand_relu (ReLU)      (None, 80, 80, 96)   0           block_1_expand_BN[0][0]
__________________________________________________________________________________________________
block_1_pad (ZeroPadding2D)     (None, 81, 81, 96)   0           block_1_expand_relu[0][0]
__________________________________________________________________________________________________
block_1_depthwise (DepthwiseCon (None, 40, 40, 96)   864         block_1_pad[0][0]
__________________________________________________________________________________________________
block_1_depthwise_BN (BatchNorm (None, 40, 40, 96)   384         block_1_depthwise[0][0]
__________________________________________________________________________________________________
block_1_depthwise_relu (ReLU)   (None, 40, 40, 96)   0           block_1_depthwise_BN[0][0]
__________________________________________________________________________________________________
block_1_project (Conv2D)        (None, 40, 40, 24)   2304        block_1_depthwise_relu[0][0]
__________________________________________________________________________________________________
block_1_project_BN (BatchNormal (None, 40, 40, 24)   96          block_1_project[0][0]
__________________________________________________________________________________________________
block_2_expand (Conv2D)         (None, 40, 40, 144)  3456        block_1_project_BN[0][0]
__________________________________________________________________________________________________
block_2_expand_BN (BatchNormali (None, 40, 40, 144)  576         block_2_expand[0][0]
__________________________________________________________________________________________________
block_2_expand_relu (ReLU)      (None, 40, 40, 144)  0           block_2_expand_BN[0][0]
__________________________________________________________________________________________________
block_2_depthwise (DepthwiseCon (None, 40, 40, 144)  1296        block_2_expand_relu[0][0]
__________________________________________________________________________________________________
block_2_depthwise_BN (BatchNorm (None, 40, 40, 144)  576         block_2_depthwise[0][0]
__________________________________________________________________________________________________
block_2_depthwise_relu (ReLU)   (None, 40, 40, 144)  0           block_2_depthwise_BN[0][0]
__________________________________________________________________________________________________
block_2_project (Conv2D)        (None, 40, 40, 24)   3456        block_2_depthwise_relu[0][0]
__________________________________________________________________________________________________
block_2_project_BN (BatchNormal (None, 40, 40, 24)   96          block_2_project[0][0]
__________________________________________________________________________________________________
block_2_add (Add)               (None, 40, 40, 24)   0           block_1_project_BN[0][0]
                                                                 block_2_project_BN[0][0]
__________________________________________________________________________________________________
block_3_expand (Conv2D)         (None, 40, 40, 144)  3456        block_2_add[0][0]
__________________________________________________________________________________________________
block_3_expand_BN (BatchNormali (None, 40, 40, 144)  576         block_3_expand[0][0]
__________________________________________________________________________________________________
block_3_expand_relu (ReLU)      (None, 40, 40, 144)  0           block_3_expand_BN[0][0]
__________________________________________________________________________________________________
block_3_pad (ZeroPadding2D)     (None, 41, 41, 144)  0           block_3_expand_relu[0][0]
__________________________________________________________________________________________________
block_3_depthwise (DepthwiseCon (None, 20, 20, 144)  1296        block_3_pad[0][0]
__________________________________________________________________________________________________
block_3_depthwise_BN (BatchNorm (None, 20, 20, 144)  576         block_3_depthwise[0][0]
__________________________________________________________________________________________________
block_3_depthwise_relu (ReLU)   (None, 20, 20, 144)  0           block_3_depthwise_BN[0][0]
__________________________________________________________________________________________________
flatten_1 (Flatten)             (None, 57600)        0           block_3_depthwise_relu[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 57600)        0           flatten_1[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 1)            57601       dropout_1[0][0]
==================================================================================================
Total params: 80,513
Trainable params: 78,721
Non-trainable params: 1,792
__________________________________________________________________________________________________
```

Figure A.2: MobileNetV2 detailed structure including the classifier.

B
Abbreviations and Notations

Dataset and clustering acronyms

Acronym	Meaning
FER	Facial Expression Recognition
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
AU	Action Unit
ICC	Intraclass Correlation Coefficient
SGD	Stochastic Gradient Descent
ACNN	Convolutional Neural Network with Attention mechanism

C

List of Figures

D
List of Tables

E
Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[2] E. Barsoum, C. Zhang, C. Canton Ferrer, and Z. Zhang. Training deep networks for facial expression recognition with crowd-sourced label distribution. In *ACM International Conference on Multimodal Interaction (ICMI)*, 2016.

[3] E. Barsoum, C. Zhang, C. C. Ferrer, and Z. Zhang. Training deep networks for facial expression recognition with crowd-sourced label distribution. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ICMI '16, pages 279–283, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4556-9. doi: 10.1145/2993148.2993165. URL `http://doi.acm.org/10.1145/2993148.2993165`.

[4] P. Burkert, F. Trier, M. Z. Afzal, A. Dengel, and M. Liwicki. Dexpression: Deep convolutional neural network for expression recognition. *ArXiv*, abs/1509.05371, 2015.

[5] F. Chollet et al. Keras. `https://keras.io`, 2015.

[6] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989. ISSN 1435-568X. doi: 10.1007/BF02551274. URL `https://doi.org/10.1007/BF02551274`.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[8] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon. Dhall, abhinav and goecke, roland and lucey, simon and gedeon, tom. Technical report, Australian National University, TR-CS-11-02, 2011.

[9] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 2106–2112, Nov 2011. doi: 10.1109/ICCVW.2011.6130508.

[10] P. Ekman and W. V. Friesen. Facial action coding system: A technique for the measurement of facial movement. *Consulting Psychologists Press, Palo Alto*, 1978.

[11] A. Gudi, H. E. Tasli, T. M. den Uyl, and A. Maroulis. Deep learning based facs action unit occurrence and intensity estimation. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 06, pages 1–5, May 2015. doi: 10.1109/FG.2015.7284873.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016. doi: 10.1109/CVPR.2016.90.

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. URL `http://arxiv.org/abs/1704.04861`.

[15] P. Khorrami, T. Le Paine, and T. S. Huang. Do deep neural networks learn facial action units when doing expression recognition? In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015.

[16] T. Koo and M. Y Li. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of Chiropractic Medicine*, 15, 03 2016. doi: 10.1016/j.jcm.2016.02.012.

[17] S. Li and W. Deng. Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition. *IEEE Transactions on Image Processing*, 28(1):356–370, Jan 2019. doi: 10.1109/TIP.2018.2868382.

[18] P. Lucey, J. F. Cohn, T. Kanade, J. M. Saragih, Z. Ambadar, and I. A. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 94–101, 2010.

[19] A. Mollahosseini, B. Hassani, and M. H. Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *CoRR*, abs/1708.03985, 2017. URL `http://arxiv.org/abs/1708.03985`.

[20] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[21] M. Pantic, M. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. In *2005 IEEE International Conference on Multimedia and Expo*, pages 5 pp.–, July 2005. doi: 10.1109/ICME.2005.1521424.

[22] M. Pantic, M. F. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. In *Proceedings of IEEE Int'l Conf. Multimedia and Expo (ICME'05)*, pages 317–321, Amsterdam, The Netherlands, July 2005.

[23] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[24] A. Savran, N. Alyüz, H. Dibeklioglu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun. Bosphorus database for 3d face analysis. In *The First COST 2101 Workshop on Biometrics and Identity Management (BIOID)*, Roskilde University, Denmark, 7-9 May 2008.

[25] P. E. Shrout and J. L. Fleiss. Intraclass correlations: uses in assessing rater reliability. *Psychological bulletin*, 86 2:420–8, 1979.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, Jan. 2014. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=2627435.2670313`.

[27] J. M. Susskind, A. K. Anderson, and G. E. Hinton. The toronto face database. *Department of Computer Science,University of Toronto, Toronto, ON, Canada, Tech.*, Rep,2010.

[28] M. Taini, G. Zhao, S. Z. Li, and M. Pietikainen. Facial expression recognition from near-infrared video sequences. In *2008 19th International Conference on Pattern Recognition*, pages 1–4, Dec 2008. doi: 10.1109/ICPR.2008.4761697.

[29] K. Wang, X. Peng, J. Yang, D. Meng, and Y. Qiao. Region attention networks for pose and occlusion robust facial expression recognition. *arXiv e-prints*, art. arXiv:1905.04075, May 2019.

[30] Z. Yu and C. Zhang. Image based static facial expression recognition with multiple deep network learning. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, pages 435–442, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3912-4. doi: 10.1145/2818346.2830595. URL `http://doi.acm.org/10.1145/2818346.2830595`.

Declaration of Academic Integrity

I hereby declare that I have written the present work myself and did not use any sources or tools other than the ones indicated.

Datum: ....
(Signature)