ORIGINAL ARTICLE

Improvement of Feature Localization for Facial Expressions by Adding Noise

Motaz SABRI and Takio KURITA

Hiroshima University, 1-4-1 Kagamiyama, Higashi-hiroshima, Hiroshima 739-8527, Japan

Abstract: This paper investigates feature localization abilities upon injecting noise into the convolutional neural network (CNN). The proposed model intended to classify the 7 human emotional states based on facial expressions and it is shown to perform better than the earlier convolutional neural network. The internal representation of learned features emerges and a more accurate localization of those features appears when independent Gaussian noises are added to certain joints during the deep network training. We observed that the weights after the noise contaminated units lead to output that is more definite. Such behavior improves the network generalization through automatic structuration. We confirmed this by emotion classification experiments on KDEF black and Cohen-Kanade + datasets based on facial expression.

Keywords: *Machine learning, Affection computing, Additive noise, Gaussian noise, Classification, Feature localization*

1. INTRODUCTION

Beside speech and body language, emotions play an important role in our daily routines of information exchange. Interpreting emotions enhances our understanding of each other and deepens our conversations. Extending emotion understanding to computer field enriches the human computer interaction and allows more natural ways of collaboration.

Commonly, when describing an emotion, facial expressions are considered. Many algorithms rely on extracting important facial regions to predict an emotion. Facial Action Coding System (FACS) [1] is basis for many recent emotion recognition algorithms. It's a set of facial actions which defines an emotion such as contracting glabella when being angry. Such facial actions define the facial expression and therefore an emotional state [2]. In this paper, we propose a Convolutional Neural Networks (CNN) based method to classify emotions from the facial feature.

Many neural network models have been introduced to analyze facial features [3-5]. CNN's have proven to be good features extractors for such task. Their ability to observe input through different filters widens their scope of operation. They can also localize the discriminative regions for action classification instead of classifying only objects [6, 7]. Feature localization helps understanding reasons of misclassification and the way knowledge is built within the network. GoogleNet [8] is a promising architecture that relies on CNNs to extract features through

multiple perspectives. It achieves an object detection accuracy of 44% in ILSVRC challenge [9]. Such high precision has dragged researchers' attention to this architecture. The main component of GoogleNet is called the Inception model. Its architecture emulates the animal visual cortex [11]. The Inception modules are blocks that consist of Convolutional, Pooling, and RELU Layers. It can find optimal local sparse structure in a convolutional vision network due to its dual feedforward paths. We find this model constructive in our proposed neural network for facial feature extraction. To the best of our knowledge, it hasn't been used in facial feature extractions analysis before. We stacked two Inceptions blocks to extract more silent features that are used for classification.

We suppress the number of effective weights by noise injection as a regularizer. Injecting noise directs the learning algorithm towards a solution with the least possible amount of zero weights [10, 15-17]. Kurita et al. [18] analyzed the backpropagation learning behavior while injecting noise into the inputs of the hidden units during simple MLP training. In their experiment, they showed that noisy training gives the network ability to automatically structurize itself. They also observed that the links from hidden layer to output layer hold smaller weights while the links from the input layer to hidden layer holds larger weights values. Sabri et al. [17] extended Kurita's work using a deeper network model showing how the internal representations of the networks benefit from noise injection. Audhkhasi et al. [19] analyzed same technique on CNN which generates a hyperplane in noise space by maximizing noisy expectation. They showed that

injecting noise into neural network boosts the backpropagation training of a neural network.

Motivated by the need to expand neural network generalization abilities, we analyze injecting Gaussian noise selectively to multiple joints within a deep CNN and its relation with feature localization and performance. This is conducted through emotion classification experiments based on facial expressions. We generate activation maps and feature maps to visualize the internal representation of the network. Our experiments steadily result in accuracy improvements and clearer localization abilities of important features (e.g. higher activation around happiness facial features (Mouth and cheeks) and stressing facial features (glabella)) upon injecting Gaussian noise into selected joints of the neural network to classify the 7-emotional states in the KDEF dataset [20] and CK+ dataset [21]. Our experiments used a stacked Inception models of GoogleNet [8] that we connected to a classifier to classify KDEF portrait images into 7-emotional states. This matches the finding of recent researches that evaluate internal representations change upon regularizing the training of a network [17]. Critical facial parts have higher activations while other regions have lower values when compared to the noiseless case.

This paper is organized as follows. In Section 2, relevant models and noise injection techniques are reviewed as related work. The details of the approach used in this paper are explained in Section 3. The model definition for noiseless and noisy training is mentioned in Section 3.1 and 3.2 respectively. Feature visualization details for both feature maps and class activation maps are mentioned in Section 3.3. Experiments details are described in Section 4. Discussion of the results is in Section 5. Finally, the conclusion is shown in Section 6.

2. RELATED WORK

CNN's efficiency in various recognition tasks has dragged many research attentions. Many structures have been used in the field of emotion classification [13,14]. Facial expressions are considered key as elements when analyzing an emotional state but a challenge as well due to the variety of sizes of facial components for different subjects. In this work, we propose CNN architecture inspired by GoogleNet Inception model that classifies emotional states based on facial expressions. We also show that injecting noise during the training improves the network classification and feature localization abilities. Here, we discuss the work most related to this paper: Inception model, regularizing neural networks by injecting noise.

2.1 Inception model

Szegedy et al. [8] proposed Inception model as part of GoogleNet architecture. It consists of Convolutional, Pooling, and RELU Layers. This architecture abstracts features from different scales and allows dimension reduction before convolving over filters output with a large patch size. Inception modules optimize sparse structure through searching for a local minimum in certain domains. The Inception block first reduces the dimension since it convolves with a kernel of small size. Rectified linear unit (RELU) is used to enforce generalization and suppress ambiguity. The RELU output is then forwarded to convolution layer with bigger kernel size. The same data that was given to the Inception modules has its dimension reduced by max pooling and then forward to convolution layer. The different kernel sizes support multiple analysis scale and results in numerous output filter banks that is concatenated into a single vector forming Inception output.

Those characteristics allowed many researchers to use Inception model in classification problems for features with different sizes. This is found to be essential when it comes to facial expression interpretation. Regions like lips, eye corners and eyebrows shapes are key local regions with high sparse representations. Sun et al. [23] used Inception model in face recognition using stacked convolution and Inception layers. They recognized subjects faces regardless the distance of the subject from the camera. Schroff et al. [24] used Inception model to learn mapping from face images to a compact Euclidean space to find similarities considering multiple subject facial alignments. Their method optimizes the features themselves instead of the intermediate bottleneck layer. Introducing the parallel paths inspired by inspection model overcomes size diversity problem of the traced features. A similar scheme was proposed for people age and gender classification [25].

2.2 Regularization: Noise injection

The nature of data has been always a challenge when training CNN's. Altering the training process of a deep neural network improves the performance of learning process via numerous techniques [26]. Gaussian Noise injection to hidden units has been utilized in many neural networks researches for many years [12,27,28]. Kurita et al. [18] added noise into the hidden layers of an MLP resulting in automatic structurization which improves the generalization ability of the network. Adding randomness improves the ability to escape local minima or passing through the early training phase in the most optimum phase. Zeyer et al. [28] showed that adding gradient noise avoids over-fitting and result in lower

training loss. Sabri et al. [17] extended Kurita et al. [18] work by analyzing the performance of deeper network structures upon injecting noise into certain locations. Their extended mathematical and experimental work matches the findings of Kurita et al. [18].

2.3 Emotion recognition by using CNN

In this paper, we extend the usage of Inception model towards emotion analysis. Recent CNN architectures offer significant improvements in understanding facial expressions. Song et al. [29] utilized a deep CNN for learning facial expressions. They used multiple layers of CNN and local filter layers and achieved an accuracy of 99.2% on the Extended Cohn-Kanade Dataset [30]. When using similar architectures with a more challenging dataset, classification accuracy becomes lower. Kukla et al. [32] used two CNN models to classify emotions in KDEF [20] dataset. Their work is represented in a confusion matrix with values vary between 54% and 92%. For the same dataset Ruiz-Garcia et al. [31] using cascade networks having results confusion matrix with values between 76% and 94%. Kuilenburg et al. [33] showed results between 80% and 97% for the same dataset. Garcia et al. [22] compared the performance of Support Vector Machines (SVM) and a Multilayer Perceptron Network (MLP) on facial expression classification. They cropped irrelevant spatial features and applied Gabor filters to images as pre-processing step before classification. Their SVM proposed model with accuracy rate of 97.08% overcomes MLP approach which has an accuracy rate of 93.5%.

It is interesting to understand the nature of features extraction that leads to those results. Visualizing how a network uses the extracted features to build knowledge is key to understand why misclassification occurs. Visualization can be for network weights or attention regions. Ren et al. [34] provide insights about the region-wise visualization. They showed that the position of convolution sliding window provides localization information regarding the image. This resulted in maps that they refer

to as feature maps. Zhou et al. [6] proposed a method that allows classification-trained CNNs to learn to perform object localization, without using any bounding box annotations. They refer to it as Class Activation Mapping (CAM). CAMs allow visualizing the predicted class scores on any given image, allowing the objects detected by the CNN to be highlighted.

3. PROPOSED METHOD

In this paper, we evaluate the effect of injecting noise into multiple joints of CNN connected to a classifier. Our network is trained to classify emotions based on facial expressions as features, the emotions are happiness, surprise, neutral, fear, sadness, disgust and anger. The input data passes through 2 Inception blocks. Each block learns detailed and useful features of the input data, in our case, the KDEF [20] and CK+ [21] databases of people who show emotions through facial expressions [20]. The second block is connected to a classifier that decides which class each input data belongs to depending on a teacher signal.

In order to enable such architecture to realize the diversity of the input, multiple filters can be applied to the input. The filters are realized by sharing weights of neighboring neurons. Comparing with MLP training, CNNs require fewer weights updates during training, since multiple weights are bound together.

3.1 Proposed network architecture

Faces that hold the same emotional expressions are classified within the same category based on special features within them. The proposed deep CNN architecture in Figure 1 has two main blocks stacked and connected to a classifier. The input images are gray scaled in the preprocessing stage. The input is then tunneled to filters of Convolution 1 which analyzes the local regions of the input image and groups into filter banks. We end up with a lot of clusters concentrated in a single region. The filter's output is then forwarded to the max pooling which down-samples the images and then

Figure 1: The proposed architecture shows the two Inception blocks stacked and connected to a SoftMax classifier, the kernel size is shown on every block and S represents the Stride and P represents the Padding.

they are normalized. The normalized output is then forwarded to two stacked Inception blocks. Each block extracts more silent features through its parallel path. The detected silent features are forwarded to a classifier to classify them based on a SoftMax loss function.

The structure we used assumes the data is initially gray-scaled in a preprocessing stage. The data enters 9×9 kernel convolutions with a stride of 2 and padding of 3 followed by a 2×2 max pooling with stride and padding of 1 to down-sample the 64 different filters resulted from the previous convolution. The output is later normalized. The normalized data is then forwarded to a 1×1 convolution kernel with a stride of 1 and padding of 1 followed by 5×5 convolution kernel with a stride of 1 and padding of 1 in the upper path. The upper path reduces the dimension of the input due to the presence of 1×1 convolution. At the same time in a parallel path, the normalized data enters a 5×5 max pooling followed by a 1×1 convolution kernel with a stride of 1 and padding of 1. Both outputs are later concatenated. This concatenation offers more diverse representation of the input. Another identical Inception block receives the concatenated data and its output is forwarded into a 4×4 max pooling with stride of 2 and padding of 1. The max pooling output is forwarded to a 7-classes classifier with SoftMax loss function.

As proposed by GoogleNet [8], kernel sizes are recommended to be between 1×1 , 3×3 and 5×5 . In our work, our selection for kernel sizes reflects the various scales at which facial expressions can appear.

3.2 Training with noise

Introducing noise during training showed an improvement in classification tasks [17,18]. We believe adding noise to the activations of Inception modules hidden layers will improve accuracy and reduce the loss. Unlike standard neural network, convolutional neural network components are more sensitive to noise. Injecting noise into Max pooling or batch normalization units, for instance, disturbs the sole purpose of those units and such interference is considered destructive. In our case, the output of every convolution unit is a candidate for injection. The smallest kernel results in more filters. Therefore, injecting noise at the 1×1 convolutional unit output gives the highest impact.

Contaminating filters output results in similar effects as standard neural network weights contamination. As proposed in [17,18] learning rate is suppressed when network is uncertain about the current output. This gives less value for unconstructive features. On the other hand, the learning rate is accelerated when the output is more

definite. It's also predicted that internal representation of learned features will emerge and sparsity of filter outputs will increase when independent Gaussian noises are added during the deep network training. This is expected to improve the generalization ability as well of the network through this automatic structuration by adding the noises.

In the next section, we show two schemes for noise injection. Injection at the first Inception block and at the second block. Inception models learn features depending on their location at the network. First Inception block learns features that are called first-order features. The second block can understand more features detected from the first block such as which alignment of facial features or facial features that tend to occur together to form an expression.

In our first noise placement, we add noise between $Conv2_A$ and $Conv2_B$ shown in Figure 1. The second was conducted while adding noise between $Conv3_A$ and $Conv3_B$. We evaluated the overall improvement through network classification accuracy, loss and ability of the network to localize facial features more precisely. Increasing the accuracy means the network is looking with more attention towards important features. This means feature locality is to increase consequently due to accuracy improvement due to noise injection.

3.3 Visualizing features

Visualizing the predicted class scores for network inputs highlights the discriminative object parts detected by the CNN which gives reasons for accurate or inaccurate classification. We use feature maps and class activation maps to localize deep features that offer understanding of discrimination used by CNNs. We visualized each layer results in feature vectors that come with different characteristics. We believe that noise encourages the network to identify the extent of the object as compared to the noiseless case which encourages it to identify just one generalization of the target. The learning rate is suppressed when the output has uncertain value and is accelerated when it has a definite value. We verify this experimentally on KDEF and CK+ datasets in experiment section. While noisy training achieves slightly outperforms classification performance of noiseless case, noisy case significantly outperforms noiseless for localization.

It's worth mentioning that the class activation visualizes certain class at a time. Let's say we are visualizing emotional state E , visualizing the activations of the units that leads to classifier E shows significantly higher values while other class activations converge to zero. For illustration, in Figure 2 we visualize activation maps for neutral and anger. Class activation map localizes the informative regions for

those classes. Activations with low values (>0.015) are ignored during the visualizing as suggested on Zhou et al. [6]. The network is focusing on the eyes when giving the decision about neutrality while it looks both at eyes and mouth (specifically teeth presence) when classifying anger. More schemes are discussed in the experiments section.

We visualize feature maps of the weights that connect the last layer classifier with the convolutional units ($MaxPool3_B$) as shown in Figure 3. In convolutional networks, a window with the filter size slides across the input searching for salience features. That way we can find features in that window. Feature maps gradually capture higher level features that consist of lower level features. Convolution units result in many filter outputs. In every case, we visualize unit outputs with the highest activations. Those units are the units to make the decision later by supplying those high activations to the classifiers.

Figure 3 shows feature maps for some emotion states without noise injection for illustration. In Figure 3(A), the network observes the lips and eye shapes to detect fear. Higher weights in those regions indicate their importance. In Figure 3(B), for instance, the eyebrows have higher weights than any other region which gives an indication of its importance when classifying anger. More schemes are considered and discussed in the experiments section.

Figure 2: This figure shows the different emotions class activation maps of KDEF and CK+ datasets. The emotions are given in the following order: (A) Neutral from KDEF and (B) Anger from CK+. The blue region indicates higher activation which means more attention is given by the network to make classification decision.

Figure 3: This figure shows the different emotions feature maps for samples of KDEF and CK+ datasets. The emotions are given in the following order: (A) Surprise (CK+), (B) Anger (KDEF), (C) Fear (CK+) and (D) Neutral (KDEF).

4. EXPERIMENTS

In our experiment, we train and evaluate classification network using KDEF [20] and CK+ [21] datasets. We evaluate the effect of noise injection during training. Beside accuracy as an important measure of evaluation, we analyze feature localization. Extracted Features for happiness are different from features for sadness. We propose two classification schemes to clearly visualize the noise effect on feature localization; A Single emotion expert and multiple emotions expert. In the single emotion expert, the classifier size is 2 (emotion E and others). In the multiple emotions expert case, the classifier size is 7 to include all emotional states. The single emotion expert classifies a single emotion state and all other states (e.g. Happiness and all others). Our classification accuracies for KDEF without noise vary between 93.0% to 98.95% and losses of 0.0012 to 0.0009 depending on the emotional state. The multiple emotions expert detection rate was 89.73% while the loss was 0.013 without noise. Classification accuracy For CK+ single emotion expert without noise varies between 93.25% to 99.08% and losses of 0.0005 to 0.0002 depending on the emotional state. The multiple emotions expert detection rate for CK+ was 92.65% while the loss was 0.0008 without noise. Experiments details and results are shown below. Analysis of the findings is found in the discussion section.

4.1 Dataset

We used KDEF and CK+ datasets in our experiments. The KDEF dataset has a population of 35 females and 35 males between 20 and 30 years of age. No beards, mustaches, earrings or eyeglasses, and no visible makeup. Subjects evoked the emotion that was to be expressed while seated three meters from the camera. The dataset comes with different angles describing 7 emotional states.

The CK+ facial expression database includes 486 sequences from 97 posers. Every sequence begins with a neutral expression and end up with one of the 6 expressions. The apex and offset of sequences are used. For both datasets, the considered emotions are happiness, surprise, neutral, fear, sadness, disgust and anger. We specifically analyzed the improvement in localization with noise injection for portrait pictures. In our experiment we used 1960 portrait images from KDEF dataset and 900 portrait images for CK+ dataset. Figures 4 and 5 show sample data. KDEF Dataset comes with a variety of the expressions depth, making it one of the challenging classification subjects [20]. The idiosyncratic hit rate was given by authors. It is defined by the percentage of correct emotion prediction by

Figure 4: This figure shows the different emotions given by KDEF. The seven emotional states are shown in random order. The states are happiness, surprise, fear, sadness, disgust, anger and neutral.

Figure 5: CK+ emotional dataset contains mixture of colored and gray-scaled emotional transitions.

an observer. The mean success index was 71.87%, ranging over the different emotions from 43.03% to 92.65% [20].

4.2 Single emotion expert case

Our network receives Images with 224×224 pixels which defines the input size of the network. We consider the network accuracy and overall loss beside the localization ability for every case. The location of the feature and the high activations around important regions are visualized. We also evaluate the training loss and testing accuracy of our model. The training took 100 Epochs with a mini-batch size of 30. The loss function for noisy and noiseless training is the SoftMax cross entropy.

In this experiment, we used 130 images of class E of KDEF dataset and 800 images of other classes. We used 1000 images for testing. We used SGD optimizer with learning rate 0.0001 without weight decays. We inject the Gaussian noise into the Inception model blocks. The noise we used comes with a mean of 0 and the standard deviation $\sigma = 0.1$ based on multiple experiments. The noise was injected during training and evaluated during testing through the accuracy of classification, loss performance and localization abilities. We conducted two experiments by varying the injection point. The first was conducted by adding noise between $Conv2_A$ and $Conv2_B$. The second was conducted while adding noise between $Conv3_A$ and $Conv3_B$.

Table 1: The results of our experiments on KDEF dataset for Emotion vs. all other emotion states is shown beside Ruiz-Garcia et al. [31] work denoted by DLE, Kukla et al. [32] work denoted as CAS and Garcia et al. [22] work denoted as SVM. Each classification task is done independently. Happiness vs. Others is abbreviated as HA, Surprise vs. Others is abbreviated as SU, Neutral vs. Others is abbreviated as NE, Fear vs. Others is abbreviated as FE, Sadness vs. Others is abbreviated as SA, Disgust vs. Others is abbreviated as DI, Anger vs. Others is abbreviated as AN.

E	SIC			DLE	CAS	SVM
	No Noise	1 st ICP Noise	2 nd ICP Noise			
HA	97.8%	98.1%	97.9%	86.3%	80.83%	100%
SU	96.3%	96.5%	96.32%	92.5%	78.33%	95.24%
NE	98.95%	98.90%	98.99%	81.5%	80.12%	100%
FE	93.0%	93.04%	92.8%	81.03%	82.5%	97.80%
SA	95.2%	96.21%	95.0%	91.3%	78.33%	96.47%
DI	94.6%	95.16%	94.69%	88.7%	70.0%	98.97%
AN	96.8%	97.2%	97.1%	85.6%	88.33%	91.36%

Classification results for noiseless scheme beside different noise injection positions are shown in Table 1. We refer to our stacked Inception classifier as SIC. The cascade use of neural networks for facial expression recognition given by Kukla et al. [32] is referred to as CAS in this paper. We refer to the Deep learning approach given Ruiz-Garcia et al. [31] as DLE. Finally, we refer to the model given by Garcia et al. [22] as SVM. CAS, SVM and DLE works are considered the state of the art when considering KDEF dataset for facial expression analysis.

The network benefits from injecting noise to the first block more than the second block. The margin of improvement between the two schemes is minor with a value of $\pm 0.2\%$. We discuss this phenomenon in the next section. SVM model shows the highest accuracies in most of the cases. Compared to SVM model, our model shows higher accuracies for anger vs. others with 97.2% and 97.1% for the first block and second block noise injections respectively. It also shows better accuracies for surprise vs. others with 96.5% and 96.32% for the first block and second block noise injections respectively. Within our model, the lowest case is for fear with 93.04% and 92.8% for first block and second block noise injections respectively. It's worth mentioning that dual injection for both joints at the same time didn't improve the accuracy more than injecting noise at the first block scheme.

For the CK+ experiment, we used 100 images of class E and 500 images of other classes. We used 300 images for testing. Same KDEF experimental setup is used. Classification results for noiseless scheme beside different noise injection positions for CK+ are shown in Table 2. Similar to KDEF findings, the network benefits from injecting noise to the first block the most. Best accuracies for neutral vs. others with 99.52% and 99.10% for the first block and second block noise injections respectively.

Table 2: The results of our experiments on CK+ dataset for Emotion Vs. all other emotion states is shown. Happiness vs. Others is abbreviated as HA-Others, Surprise vs. Others is abbreviated as SU-Others, Neutral vs. Others is abbreviated as NE-Others, Fear vs. Others is abbreviated as SA-Others, Disgust vs. Others is abbreviated as DI-Others, Anger vs. Others is abbreviated as AN-Others.

Emotion	No Noise	1 st block noise	2 nd block noise
HA-Others	98.02%	98.63%	98.22%
SU-Others	97.01%	97.86%	96.63%
NE-Others	99.08%	99.52%	99.10%
FE-Others	93.25%	95.11%	94.24%
SA-Others	96.01%	96.82%	96.21%
DI-Others	95.04%	95.16%	94.11%
AN-Others	97.82%	97.91%	97.85%

The lowest accuracy is for fear with 95.11% and 94.24% for first block and second block noise injections respectively. This shows that the advantage of injecting noise is independent from characteristics of the dataset.

In Figures 6 and 7, we visualize the feature maps and the class activation for noiseless and multiple noisy trainings. We observe stronger feature representation when noise is injected into network whether the injection is at first or second Inception block. Figure 6 shows how the internal representation of the features improves due to noise injection for both KDEF and CK+. Figure 6(A), for instance, is evaluating fear. The shape of the lips is

Figure 6: This figure shows the different emotions feature maps of KDEF and CK+ datasets. The emotions are given in the following order: (A) Contempt (CK+), (B) Disgust (CK+), (C) Fear (KDEF), (D) Anger (KDEF) and (E) Neutral (KDEF). Saliency is key when determining the efficiency of a feature map representation over the other. First block noise column is showing best feature representations.

dominating the decision-making process with significant higher weights upon injecting noise. Figure 6(C), on the other hand, is analyzing anger. After injecting noise the eyebrows region gets higher weights. In all cases, the third column shows best representations.

On the same hand, the localization abilities of the network significantly become restricted towards more critical regions after injecting noise. This is observed via class activation mapping in Figure 7 for both KDEF and CK+. Considering surprise emotion for instance in Figure 7(A), the network observed almost the entire face to give its prediction. After injecting noise, the region has got smaller. This is a significant improvement considering the accuracy has improved as well. Each emotional state benefits from the noise based on its local characteristics.

Figure 7: This figure shows the different emotions class activation maps of KDEF and CK+ datasets. The emotions are given in the following order: (A) Surprise (CK+), (B) Anger (KDEF), (C) Neutral (KDEF), (D) Fear (KDEF), (E) Happiness (KDEF), (F) Disgust (KDEF) and (G) sadness (CK+). Features locality gives an indication which regions the network depends on for decision making.

Anger classification focuses on the teeth and eyebrows. Disgust classification focuses on the mouth and eyes opening. Sadness classification focuses slightly more on the lips shape and so on. Once again, the noise injection at the first Inception block (Third column in Figure 7) benefits feature localization for all emotions.

4.3 Multiple emotions expert case

In this experiment, we used 130 images per class for training and 1000 images for testing from KDEF dataset. We utilized SGD optimizer with learning rate 0.0001 without weight decays. The proposed architecture has proven to be very effective on this dataset with an average accuracy of 90.6%. We inject the Gaussian noise into the Inception model blocks. Similar to the previous experiment, we varied the injection points. The first was conducted by adding noise between *Conv2_A* and *Conv2_B*. The second was conducted while adding noise between *Conv3_A* and *Conv3_B*. The confusion matrix in the Table 3 shows results for the noiseless training case. Confusion matrix at Table 4 shows results of the first Inception block noise injection. Table 5 shows results for injecting noise at the second Inception block.

Table 3: The confusion matrix on the KDEF Dataset is shown without injecting noise into the network. The lowest accuracy is achieved by the emotion fear with 83% while happiness is detected with 95.4%. Happiness is abbreviated as HA, Surprise as SU, Neutral as NE, Fear as FE, Sadness as SA, Disgust as DI and Anger as AN.

	HA	SU	NE	FE	SA	DI	AN	
HA	0.954	0.015	0.01	0.0	0.006	0.0	0.015	1.0
SU	0.013	0.951	0.00094	0.011	0.016	0.0006	0.00746	
NE	0.011	0.034	0.914	0.00311	0.0009	0.014	0.02299	0.5
FE	0.008	0.015	0.0295	0.83	0.025	0.0551	0.0374	
SA	0.00059	0.0405	0.022	0.03	0.8799	0.0025	0.02451	
DI	0.023	0.015	0.021	0.039	0.032	0.831	0.039	
AN	0.021	0.023	0.016	0.016	0.001	0.002	0.921	0

Table 4: The confusion matrix on the KDEF Dataset is shown while injecting noise into the first Inception model during training. The lowest accuracy is achieved by the emotion fear with 88.7% while happiness is detected with 96.7%.

	HA	SU	NE	FE	SA	DI	AN	
HA	0.965	0.015	0.02	0.0	0.0	0.0	0.0	1.0
SU	0.01	0.967	0.00094	0.009	0.005	0.0006	0.00746	
NE	0.009	0.03	0.942	0.002	0.011	0.006	0.0	0.5
FE	0.007	0.01	0.02	0.8989	0.015	0.04	0.0091	
SA	0.001	0.04	0.021	0.03	0.9035	0.001	0.0035	
DI	0.019	0.012	0.001	0.032	0.012	0.887	0.037	
AN	0.02	0.02	0.012	0.015	0.002	0.001	0.93	0

Table 5: The confusion matrix on the KDEF Dataset is shown upon injecting noise into the second Inception model. The lowest accuracy is achieved by the emotion fear with 84.11% while happiness is detected with 96.0%.

	HA	SU	NE	FE	SA	DI	AN	
Ha	0.96	0.017	0.023	0.0	0.0	0.0	0.0	1.0
Su	0.01	0.956	0.0104	0.02	0.003	0.0006	0.0	
Ne	0.015	0.03	0.925	0.005	0.02	0.005	0.0	0.5
Fe	0.015	0.03	0.03	0.8411	0.02	0.04	0.0239	
Sa	0.011	0.049	0.021	0.031	0.8835	0.001	0.0035	
Di	0.0592	0.02	0.002	0.035	0.014	0.8528	0.017	
An	0.020	0.0214	0.017	0.015	0.002	0.001	0.9236	0

We used the same experimental setup for CK+ dataset. We used 100 images per class for training and 200 images for testing. The experiment resulted in average accuracy of 90.6%. We inject the Gaussian noise into the Inception model blocks. Similar to the previous experiment, we varied the injection points. The confusion matrix in the Table 6 shows results for the noiseless training case. Confusion matrices at Tables 7 and 8 show results for injecting noise at first and second Inception blocks respectively.

Finally and to confirm the improvement in accuracy and loss after injecting noise. Figure 9 shows the training accuracy in both noisy and noiseless cases while Figure 8

Table 6: The confusion matrix on the CK+ Dataset is shown without injecting news into the network. The average accuracy is 92.65%.

	HA	SU	NE	FE	SA	DI	AN	
HA	0.963	0.005	0.02	0.0	0.0	0.01	0.002	1.0
SU	0.01	0.9611	0.00094	0.009	0.005	0.0006	0.00746	
NE	0.009	0.03	0.942	0.002	0.011	0.006	0.0	0.5
FE	0.007	0.01	0.02	0.8989	0.015	0.04	0.0091	
SA	0.001	0.04	0.021	0.03	0.9035	0.001	0.0035	
DI	0.019	0.012	0.001	0.032	0.012	0.887	0.037	
AN	0.02	0.02	0.012	0.015	0.002	0.001	0.93	0

Table 7: The confusion matrix on the CK+ Dataset is shown while injecting noise into the first Inception model during training. The average accuracy is 94.34%.

	HA	SU	NE	FE	SA	DI	AN	
HA	0.982	0.01	0.008	0.0	0.0	0.0	0.0	1.0
SU	0.001	0.971	0.0184	0.0054	0.0011	0.0001	0.003	
NE	0.0013	0.03	0.961	0.003	0.003	0.002	0.0	0.5
FE	0.011	0.03	0.03	0.9021	0.0048	0.021	0.0011	
SA	0.0032	0.004	0.017	0.037	0.937	0.0001	0.002	
DI	0.0069	0.04	0.005	0.017	0.003	0.911	0.017	
AN	0.002	0.034	0.007	0.014	0.002	0.0001	0.94	0

Table 8: The confusion matrix on the CK+ Dataset is shown upon injecting noise into the second Inception model. The average accuracy is 93.59%.

	HA	SU	NE	FE	SA	DI	AN	
HA	0.977	0.01	0.008	0.0	0.0	0.0	0.005	1.0
SU	0.0005	0.9623	0.0184	0.0094	0.0012	0.0051	0.0031	
NE	0.006	0.0239	0.9511	0.007	0.004	0.0026	0.0	0.5
FE	0.031	0.023	0.003	0.9	0.0049	0.026	0.0121	
SA	0.0038	0.0	0.016	0.0573	0.921	0.0001	0.0018	
DI	0.007	0.04	0.005	0.017	0.013	0.901	0.017	
AN	0.002	0.0344	0.007	0.0144	0.0021	0.0011	0.939	0

Figure 8: Chart shows the network loss during KDEF training and testing for the noisy and noiseless training. It's observable that the loss is having less value much faster in the noisy case. The varying between training and testing curves is natural due to the learning tendencies for the SGD optimizer.

Improvement of Feature Localization for Facial Expressions by Adding Noise

Figure 9: KDEF Accuracy chart shows the improvement of the network learning through epochs. With the two cases, we trained 100 epochs with and without using noise. The noisy case shows observable improvement over the noiseless case.

Figure 10: Classification loss is shown during training and testing of CK+ for the noisy and noiseless training.

Figure 11: The classification accuracy of CK+ shows the improvement through epochs.

shows the loss in both cases for 100 epochs training for KDEF dataset. Figures 11 and 10 show the accuracy and loss for both noisy and noiseless case for CK+ dataset.

5. DISCUSSION

The accuracy of classification on the KDEF and CK+ datasets shows that the proposed architecture is robust and comparable to the state of the art methods. Our method shows better performance against Ruiz-Garcia et al. [31] and Kukla et al. [32] approaches. Compared to our work, the SVM model proposed by Garcia et al. [22] results in better detection accuracies for 5 out of 7 emotion classification. It's worth mentioning that the preprocessing phase of their work is unavoidable. The face detection

and Gabor filtering are expensive tasks considering real-time processing. The training time in our work is time consuming. The produced model processes videos frames in a real time manner offering a comparable accuracy.

Misclassification usually occurs because the network is looking at undesirable regions of the image. Some undesirable regions won't necessarily cause misclassification but they definitely affect the accuracy of the neural network on the long run. Initially, without any interference with the training, the network starts observing regions within subject's faces that are critical even for humans to give decisions about an emotional state. Those regions, as we illustrated in Figure 2, are the eye region, nose and mouth. Later after injecting noise, those regions become more restricted to more important areas. Regions like eye corners, laugh lines beside the nose or marionette lines below the lips get higher importance. The darker the color of the activation map, the higher the activation. Our network detected the importance of those regions without any further training on where to look.

To determine the best location for noise to be injected into our model, we consider accuracy as a measure. From Tables 5 and 7, accuracy is the highest when noise is injected in the first block offering special features for the training that was given to the second block. To understand why the second injection scheme isn't as efficient as the first injection scheme we can observe the network visualization. Many of the activation maps are showing disturbance on their outputs that was delivered to the classifier which results in less accuracy. We think this is because the network didn't benefit long enough from the noise and was directly tunneled to the classifier which has many filter outputs to be processed. This resulted in minor improvement in accuracy and localization but wasn't as powerful as the previous injection scheme. This also clarifies why injecting noise into both (first and second block) points does not benefit the network performance to be better.

The feature localization doesn't only tell us where the network is looking to make a decision but also how important is that region in making this decision. In Figure 7, facial expressions of anger, fear and surprise reside in the same region which is mainly around the mouth. Many researches indicated that those 3 emotional states tend to be confused with each other by machine learning algorithms because of their close locality. Our activation maps are showing unique visualization for every state which justifies why accuracies in confusion matrix in Tables 4, 5, 7 and 8 are better than confusion matrix in Tables 3 and 6.

The network shapes its representation gradually to give an approximation of the input. The feature map visualization shows a weak representation of the input without noise. The results are good in such case but once the noise is injected the features become more salient. Injecting noise at the first block benefits the network into building up a structured representation which is used to train the second block giving an accurate feature map as shown in the first block noise column in Figure 6. When the noise is injected into the second block, the network uses it to improve the already existed representation and didn't benefit much as shown in Figure 6. Feature maps show predictable behavior as mentioned in [17] when noise is injected. It achieves the ability to generalize the network decisions through automatic structuration.

6. CONCLUSION

In this paper, we investigated the effect of injecting noise on enhancing localization of facial features and improving generalization of CNN made of GoogleNet Inception blocks and a classifier. We proposed detailed explanation through feature map and class activation maps representation on why injecting noise at certain locations increases the network performance. Experimentally, injecting noise at those locations offers better performance for both deep and shallow networks. Trained CNNs has learned to perform object localization and highlighting the discriminative object parts, without using any bounding box annotations.

GoogleNet Inception module produces generic localizable deep features. Combining this property with noise injection opens high research potential ahead in investing this design space to enhance existing networks performance and generalization ability.

ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI Grant number 16K00239 and 16H01430.

REFERENCES

- Ekman P., Friesen W.V., Hager J.C.; Facial Action Coding System, Research Nexus, Salt Lake City, 2002.
- B.V. Kumar; Face expression recognition and analysis: the state of the art, Course Paper, Visual Interfaces to Computer, 2009.
- E. Kim, and S. Vangala; Deep Action Unit classification using a binned intensity loss and semantic context model, 23rd International Conference on Pattern Recognition (ICPR), pp.4136-4141, 2016.
- P. Khorrami, T.L. Paine, and T.S. Huang; Do Deep Neural Networks Learn Facial Action Units When Doing Expression Recognition, International Conference on Computer Vision Workshop, pp.19-27, 2015.
- S. Ghosh, E. Laksana, S. Scherer, and L.-P. Morency; A Multi-label Convolutional Neural Network Approach to Cross-Domain Action Unit Detection, In Proceedings of ACII 2015, pp.19-27, 2015.
- B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba; Learning Deep Features for Discriminative Localization, CVPR 2016, pp.2921-2929, 2016, (arXiv:1512.04150, 2015).
- M.D. Zeiler, and R. Fergus; Visualizing and understanding convolutional networks, In Computer Vision-ECCV, pp.818-833, 2014.
- C. Szegedy et al.; Going deeper with convolutions, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1-9, 2015.
- LSVRC; Results of the lsvrc challenge. [Online]. Available: <http://www.image-net.org/challenges/LSVRC/2014/results> 2014.
- C.S. Leung, H.-J. Wang, and J. Sum; On the Selection of Weight Decay Parameter for Faulty Networks, IEEE Transactions on Neural Networks, 21(8), pp.1232-1244, 2010.
- C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna; Rethinking the Inception Architecture for Computer Vision, arXiv:1512.00567, 2015.
- G. An; The effects of adding noise during backpropagation training on a generalization performance, Neural computation, pp.643-674, 1996.
- C.-D. Caeleu; Face expression recognition: A brief overview of the last decade, in Applied Computational Intelligence and Informatics (SACI), 2013 IEEE 8th International Symposium on, pp.157-161, 2013.
- V. Bettadapura; Face expression recognition and analysis: the state of the art, arXiv preprint:1203.6722, 2012.
- Y. Bengio; Deep Learning of Representations: Looking Forward, International Conference on Statistical Language and Speech Processing, pp.1-37, 2013.
- Z. Lian, X. Jing, X. Wang, H. Huang, Y. Tan, and Y. Cui; Drop Connect Regularization Method with Sparsity Constraint for Neural Networks, Chinese Journal of Electronics, 25(1), pp.152-158, 2016.
- M. Sabri, and Takio Kurita; Effect of Additive Noise for Multi-layered Perceptron with AutoEncoders,

- IEICE Transactions on Information and Systems, 7, pp.1494-1504, 2017.
18. T. Kurita, H. Asoh, S. Umeyama, S. Akaho, and A. Hosomi; A Structural Learning by Adding Independent noises to Hidden Units, Proceedings of the IEEE International Conference on Neural Networks, IEEE World Congress on Computational Intelligence, 1, pp.275-278, 1994.
 19. K. Audhkhasi, O. Osoba, and B. Kosko; Noise enhanced convolutional neural networks, Neural Networks, 78, pp.15-23, 2016.
 20. Lundqvist, D., Flykt, A., and Öhman, A.; The Karolinska Directed Emotional Faces - KDEF, CD-ROM from Department of Clinical Neuroscience, Psychology Section, Karolinska Institutet, Stockholm, 1998.
 21. Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., and Matthews, I.; The Extended Cohn-Kande Dataset (CK+): A complete facial expression dataset for action unit and emotion-specified expression, CVPRW, 2010.
 22. Ruiz-Garcia, A., Elshaw, M., Altahhan, A., and Palade, V.; Emotion Recognition Using Facial Expression Images for a Robotic Companion, In International Conference on Engineering Applications of Neural Networks, pp.79-93, 2016.
 23. Y.Y. Sun, L. Ding, X. Wang, and X. Tang; DeepID3: Face recognition with very deep neural networks, CoRR, abs/1502.00873, 2015.
 24. F. Schroff, D. Kalenichenko, and J. Philbin; FaceNet: A Unified Embedding for Face Recognition and Clustering, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.815-823, 2015.
 25. G. Levi, and T. Hassner; Age and Gender Classification using Convolutional Neural Networks, IEEE Workshop on Analysis and Modeling of Faces and Gestures (AMFG), at the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, pp.34-42, 2015.
 26. G.E. Hinton, N. Srivastava, A. Krizhevsky, I Sutskever, and R. R Salakhutdinov; Improving neural networks by preventing co-adaptation of feature detectors, 2012.
 27. R. M. Zur, Y. Jiang, L.L. Pesce, and K. Drukker; Noise injection for training artificial neural networks: A comparison with weight decay and early stopping. Med Phys., pp.4810-4818, Published online 2009.
 28. A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schlter, and H. Ney; A Comprehensive Study of Deep Bidirectional LSTM RNNs for Acoustic Modeling in Speech Recognition, arXiv:1606.06871v1, 2015.
 29. I. Song, H.-J. Kim, and P. B. Jeon; Deep learning for real-time robust facial expression recognition on a smartphone, in Consumer Electronics (ICCE), 2014 IEEE International Conference on IEEE, pp.564-567, 2014.
 30. Kanade, T., Cohn, J.F., and Tian, Y.; Comprehensive database for facial expression analysis. Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00), Grenoble, France, pp.46-53, 2000.
 31. A. Ruiz-Garcia, M. Elshaw, A. Altahhan, and V. Palade; Deep Learning for Emotion Recognition in Faces, 25th International Conference on Artificial Neural Networks, pp.38-46, 2016.
 32. Kukla, E., and Nowak, P.; Facial Emotion Recognition Based on Cascade of Neural Networks, Advances in Intelligent Systems and Computing, pp.67-78, 2015.
 33. van Kuilenburg, H., Wiering, M., and den Uyl, M.; A model based method for facial expression recognition, Lectures Notes in Computer Science, 720. Machine Learning, ECML, pp.194-205, 2005.
 34. S. Ren, K. He, R. Girshick, and J. Sun; Faster R-CNN: Towards real-time object detection with region proposal networks, In NIPS, 2015.

Motaz SABRI (Non-member)

Motaz Sabri received a master degree in computer engineering from Hiroshima University. Presently he is a PhD. student at Hiroshima University. He is enrolled in big data laboratory. His main research interests are features classification, pattern recognition and affective computing and their applications to image and video understanding via wavelets and neural networks.

Takio KURITA (Non-member)

Takio Kurita received the B.Eng. degree from Nagoya Institute of Technology and the Dr. Eng. degree from the University of Tsukuba, in 1981 and in 1993, respectively. He joined the Electro-technical Laboratory, AIST, MITI in 1981. From 1990 to 1991 he was a visiting research scientist at Institute for Information Technology, National Research Council Canada. From 2001 to 2009, he was a deputy director of Neurosciences Research Institute, National Institute of Advanced Industrial Science and Technology (AIST). Also he was a Professor at Graduate School of Systems and Information Engineering, University of Tsukuba from 2002 to 2009. He is currently a Professor at Hiroshima University. His current research interests include statistical pattern recognition and its applications to image recognition. He is a member of the IEEE, the IPSJ, the IEICE of Japan, Japanese Neural Network Society, The Japanese Society of Artificial Intelligence.