ISSN 2278-3091

Volume 9, No.1.4, 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse0291.42020.pdf

https://doi.org/10.30534/ijatcse/2020/0291.42020

Performance Analysis of Pre-Trained Residual Neural Network on Micro-Expressions Recognition

Ismail Ibrahim[1], Shuzlina Abdul-Rahman[2], Nor Hayati Abdul Hamid[3], Mohd Razif Shamsuddin[4]

[1234]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia.

aelibrahim96@gmail.com, shuzlina@tmsk.uitm.edu.my, nhayati@tmsk.uitm.edu.my, razif@tmsk.uitm.edu.my

ABSTRACT

This paper presents the performance analysis of pre-trained Residual Neural Network on micro-expression recognitions. Due to recent micro-expression grand challenge, the micro-expressions started to become a hot topic in machine learning domains. The importance of micro-expressions become vitals in searching for the clues that lead to the detection of deceit's behavior. Many researchers usually created the custom deep learning algorithm, but often the existing state-of-the-art algorithms being neglected. Thus, this paper aims to provide an analysis on the performance of the five Convolutional Neural Network's variants, specifically on Residual Neural Network. This is to find the exact point on why the current state-of-the-art models do not behave as expected based on provided datasets. These datasets were prepared in terms of training, validation, and testing dataset, in which three types of frames namely as onset, apex, offset frames were used. The datasets prepared were combined altogether from renowned resources; CASME I, CASME II, CAS(ME)^2, and SAMM datasets. For the performance results, one of the ResNet variants known as ResNet18 could only achieve up to 17% accurate with the learning rate of 1e-5 on the given datasets. This indicated that the existing models do not performed very well. Another possibility for this result is because the data itself probably hard to distinguish between two or more classes since it is a subtle facial movement. In the future, many existing models will be tested, to find the benchmark of each model on the given datasets.

Key words: Micro-expressions, ResNet, performance analysis, state-of-the-art models, Convolutional Neural Network.

1. INTRODUCTION

There are two types of expressions, known as macro and micro-expression. Macro-expression is an expression that is very explicitly toward the other persons, where these people could determine such expression exhibit by the expressors. The micro-expression is total opposite of macro-expression, where it is very implicit and could not detect through the naked eyes. In other words, micro-expression is an expression that very subtle [1].

There are several distinctive behaviors for these categories, where the macro-expressions could be exploited by the expressor. Macro-expressions could become an instrumental to falsifying the information toward the others. This usually being used to mask or hide themselves.

In contrary, micro-expressions cannot be controlled by anyone. The muscular movement of micro-expressions are usually involuntary, and it happened below than one second. Thus, it is known as spontaneous movement. As it is spontaneous movement, it being categorized as genuine expression. Such expressions usually leaving a trail of real intention of the expressors. Another take away of important key is that these expressions only appear if the owner of expressions try to suppress their current feeling [2].

Due to nature of these expressions, the ability of computer vision is needed. These is because unlike humans, the algorithms usually focus on a singular task, as an example in this case, it could use available resources to compute the facial movement and finally classifying the category of an expression.

For years, since the ILSRVC 2012, Convolutional Neural Network (CNN) became state-of-the-art algorithms that performed well in recognizing or detect an object in an image. Thus, this algorithm and its variants always being customized and used to detect micro-expressions.

Due to the nature of micro-expressions, this paper demonstrates an analysis for the current behavior of the first five CNN state-of-the-art models. The aim is to find a turning point for improving the existing algorithms.

2. LITERATURE REVIEW

2.1 Expressions

The foundation of an expression depends on the feeling of a person. While traditional acceptance is that expressions are universal throughout entire humanity [3], there are several arguments that opposed those view. In a sense, the expressions of a person are constructed throughout micro-level, where it is involving the interconnected biological neurons [4].

While Ekman is a psychological expert and being supported by Tomkins and Izard, Barrett in the other hand provided more concise arguments, as she is neurologist. Due to the fact both factions are having their own opinions and facts, backed by their experiments, facial expressions without doubt are communication instruments throughout entire homo-sapiens, due to expressions could sending an information regarding with the emotions of a person.

Micro-expressions were found by Haggard and Isaac in the year of 1966, followed by Ekman whose found the micro-expression during interview session with the psychiatric patient. He found the expression when he examined one by one of video's frame of his interview with the patient. This expression fleeting across patient's face, below than one second. Thus, this expression is known as micro-expression.

From here and now on, the micro-expressions being studied by various researchers from various scientific fields. One of the discoveries is that micro-expressions cannot be controlled by the expressor. Even though expressor tries to control it, the facial movement is symmetrical different from guanine expressions [5].

Micro-expression has the potential to become a clue for the act of deceit, which such recognitions are needed in professional domains, such as interrogation sessions at the airport by security guard or during psychiatric interview between psychiatrist and their patients. The use of micro-expressions, however, depends on environmental context.

2.2 Artificial Neural Network

There are two types of Artificial Neural Network (ANN), which known as Feed Forward Neural Network (FFNN) and Back Forward Neural Network (BFNN). The FFNN is a type where the neural network passing data from input neurons to the output neurons in forwarding manners, one direction. This means that this type of neural network does not propagating in backward manners, no feedback message throughout highest layer to lowest layer of the neural network [6]. The most popular example is the Single Layer Perceptron and Multilayer Perceptron.

The BFNN is a neural network that return message from highest layer to the lowest layer after producing loss values. By doing this, the weights connecting each neuron updated accordingly. Another feature is that this type of neural network produced coordinated graph in sequential manners. Recurrent Neural Network and Convolutional Neural Network could become as an example for this type of neural network.

Both FFNN and BFNN can extend to be a complex, deep neural network. Recurrent Neural Network (RNN) is better at classifying sequential data. Convolutional Neural Network (CNN) often used in computer vision. This is because CNN takes height and width, together with color channels as an input. Another fact that need to take note is that CNN was inspired by the discovery of Hubel and Wiesel in the year of 1962, where they found a structured nervous system of visual cortex in mammals. Inspired by this discovery, Fukushima has first proposed a neural network known as NeoCognitron, a neural network extended from Cognitron [7].

This algorithm then being extended by [8], known as Convolutional Neural Network, specifically being model as five layers, given a nickname as LeNet-5. CNN consists of local receptive fields to capture the features available in an image. Interestingly, this algorithm able to share its weight by switching it dependently during the recognition process. It also can subsample the image by reducing the dimension of the matrixes [9]-[10]. Another interesting study was conducted by Shamsudin et al. [11]. The study highlighted the generalization problem in Shallow Neural Network despite its extensive usage.

2.3 Convolutional Neural Network

However, this algorithm was not extended until a decade later, due to computational constraints. Seeing the opportunity to make it much better, [12] take a huge step extending this algorithm by utilizing multiple GPU in parallel way. He then become a runner up for ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012.

Simonyan & Zisserman [13] came up with multiple new models named as Visual Geometry Group. One of major contribution of these models are that they incorporate 3x3 filter sizes throughout networks. This means that this research leads most of the others to use uniform, smaller filter size. Another point is that this model used padding to coup with spatial problems. Figure 1 show the architecture of VGG with 11 layers. It has size of 224x224 in terms of height and width of the input image, 3x3 filter size with stride 1 through entire network, 8 convolution layers together with max-pooling, and three fully-connected layers at the end of the network.

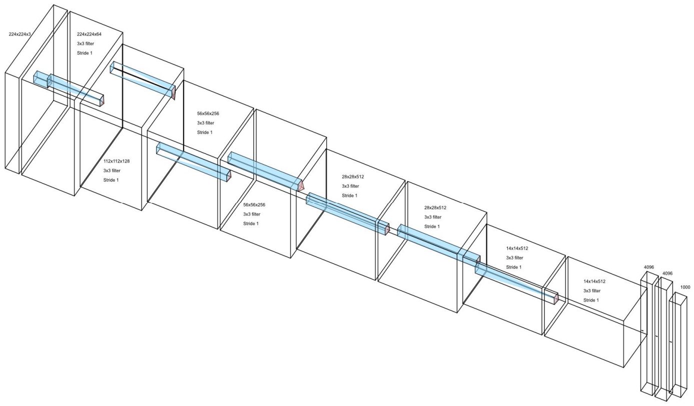


Figure 1: VGG-11 architecture by Simonyan & Zisserman (2014)

GoogleNet is the first network that used inception module. This module concatenates several features map with different size of kernel together in a building block to find spatial information. Another contribution of this model is by using 1x1 kernel size to reduce computation but reach with computation.

ResNet is one of the first family going deeper convolution, as deeper as 152 layers. It introduced the skip connection, which derived from Highway Network [14] by adding the previous input *x* and the current input *f(x)*, it can reduce the vanishing gradient's problem. Figure 2 show a part of model's architecture, thus giving an insight of how it is working.

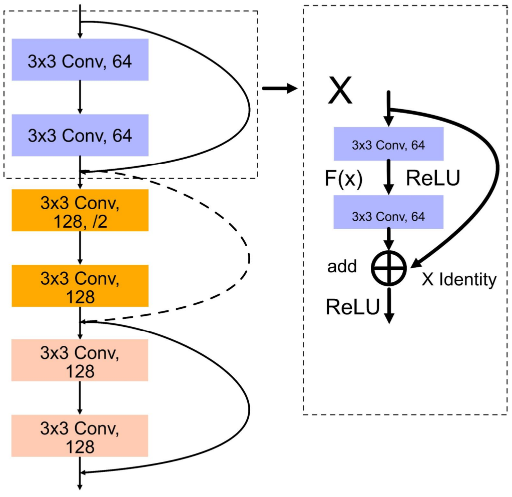


Figure 2: A part of ResNet model by He et al. (2015).

Later, the improvements on the inception modules for CNN architecture came out [15]. Before creating the spread-out convolutions, it is better to reduce the input dimension before the spatially aggregate the dimensions in lower hierarchy. This could lead to lower adverse effect after the aggregation. Finally, the optimal way to construct the CNN is by balancing the depth and width of the CNN. To do this, the computation budget must be taken into account during structuring the depth and width of the CNN.

Dense Network or known as DenseNet is a model that takes the previous input of convolution layer in terms of feed-forward fashioned. In a sense, this model could just reuse the features of old transformation data, causing it to differentiate information and such information also could be preserved. This model has shorter path of gradient's propagation, and through this design, the flow of information improves. This architectural design is based on drop-path philosophy, which is regularization method that reduce overfitting on smaller datasets during training [16].

In a paper that explained the experimentation on Squeeze Neural Network (SqueezeNet) used 1x1 kernel filters instead of 3x3 kernel filters. As they suggested, 1x1 kernel filters produced 9 times lower in terms of parameters. For the second strategy, they decrease the inputs to 3x3 kernel filters. The third strategy is downsampling the data occurs very late in a CNN to produce the large activation maps. To make the CNN algorithm robust, the fire module being introduced. The fire modules contained squeeze convolution layer that have 1x1 kernel size. The output of these layers then being inserted into expand layer that consisted 1x1 and 3x3 kernel filters [17]. For the SqueezeNet, the first layer is a convolutional layer. The convolutional layer for the next subsequent layers increased the number of filters. The zero-padding is being padded in 3x3 filters of the expand layers. The dropout with 50% is apply when the fire9 modules being exposed. Other features for SqueezeNet is that there are lower fully-connected layers. SqueezeNet used dropout as regularizer after fire9 module.

In [18], the author proposed new algorithm that became one of the CNN's variants. This model known as ResNeXt. As the authors mentioned, they adopted the topology of VGG/ResNet model. In addition, they also used the Inception design, which concatenating the output of convolution layers. In depth, they are concatenating the residual modules. It is a good strategy when combining the other model's architecture to increase accuracy.

For the same year, another ResNet variants being introduced. It is known as Wide ResNet, experimented by the author of [19]. Interesting point of view, both authors placing the dropout between the convolution layers. It worth to remember that many previous models usually implementing the dropout modules at the fully connected layers. This means that they substituted the batch normalization modules with dropout modules. Wide ResNet used 16 layers depth, which is lower than vanilla ResNet. This experiment showed that widening the ResNet layers and reducing the depth could increase the training speed.

Starting of the year 2018, many researchers started to assess the architecture of CNN that can fit into low end devices. The MobileNet-v2 was shortly introduced in [20]. In this work, the authors create a thin bottleneck that contained shortcut connections, which this design was additionally inverted. The depthwise convolutions were implemented in intermediate expansion layers for filtering features in lightweight manners. To maintain the representation of input data, the authors removed non-linearity in the narrow layers.

MnasNet was introduced later in [21]. The model was built upon on the principle of hierarchical search space factorization. This will be produced unique blocks that have separate connections. Instead of handcrafted architectural design, they used Recurrent Neural Network (RNN) as controller, a trainer, and inference engine based on mobile phones to measure the latency. As for convolution layers, they used regular, depthwise, and inverted bottleneck design in the search space. In additions, they use 3x3 and 5x5 kernel size. They used ratio of 0 and 0.25 for squeeze-and-excitation process with filter size denoted as *F*, together with number layers for a block denoted as *N*.

One of the state-of-the-art models is known as ShuffleNet-v2 being introduced [22]. In this paper, authors introduced the channel split, which the input features split into two branches during at the beginning of each units. A branch will act as identity function. Another branch will follow the principle of minimizing memory cost by using the same width of channels. This branch consisted of three convolution layers. Next, the convolution's outputs for both branches will be concatenated. To make sure both branches communicated, the channel shuffle operation was used. The depth-wise and ReLU being applied in only a branch. To follow the principle of safely ignoring the element-wise operations, they used concatenation, channel split and channel shuffle operations.

2.4 Convolutional Neural Network for Micro-Expressions

There are several methods that has being used to detect micro-expressions using CNN. The past work in [23] combined both Long Short-Term Memory (LSTM) network and CNN to extract the spatial and temporal features from the datasets to detect the micro-expression's behavior. The datasets that being prepared for this work were categorized into negative, positive, and surprise classes. There is several mini CNN to extract features from several cropped region of the faces, then being forward into LSTM to get spatial and temporal features entirely. This features then being concatenated before feeding into the fully-connected layer, and then being classified into either of these classes. The best results for this dataset are 0.9022 for Unweighted F1-score (UF1) and 0.9018 for Unweighted Average Recall (UAR).

Another work in [24] make worth with combination between Recurrent Neural Network (RNN) and CNN. However, this work suggested to capture the spatiotemporal deformation from micro-expression sequences. This spatiotemporal deformation is modeled in terms of facial appearance and geometry based separately. Appearance based method is where the one sequence of data converted into a matrix by concatenating the frames to reserved the appearance of facial regions, where geometry based method is transforming a sequence of data into a matrix by computing optical flow between onset and apex frames to get the geometric information regarding facial movement. They used LOSO and LOVO protocol to evaluate each model. Appearance based performed better in LOVO protocol, where geometry based performed better in LOSO protocol.

Some of the work incorporate state-of-the-art models, which in [25] used the Inception module to classify the data in three classes, known as negative, positive, and surprise class. In this work, the authors set up Inception module in terms of duality. The inception module used the extracted optical flow as an input data, managed to score around 0.7322 for UF1 and 0.7278 for UAR.

Another work incorporates aggregated style and attention transfer by using ResNet model as the backbone and a teacher model. This is shown in [26]. The attention of the teacher model then being transfer into student model, which managed to score 0.5936 for UF1 and 0.5958 for UAR on negative, positive, and surprise classes.

A work that used optical flow as an input can be shown in [27], where the authors introduced part-based average pooling to obtain the discriminative representation from the static face structure. Authors also introduced two domains, which known as adversarial training and expression magnification and reduction (EMR). There are two experiments being conducted, where part-based models trained with a combination of EMR and part-based models combined with EMR and adversarial training. For the first experiment the model managed to score 0.7663 for F1 and 0.7531 for UAR. The second method of training managed to score 0.7885 for F1 and 0.7824 for UAR. Both of this being conducted using full dataset (negative, positive, and surprise classes).

3. METHODOLOGY

In this experiment, there are four types of datasets being used, which known as CASME I, CASME II, CASME^2, and SAMM datasets and organized into bigger database. This is being illustrated in figure 3. The CASME datasets being collected from Chinese Academy of Science, as well as SAMM is from University of Manchester. Each image being recorded in provided excel files, where this file recorded the subject directories, the emotions, and the name of the important files.

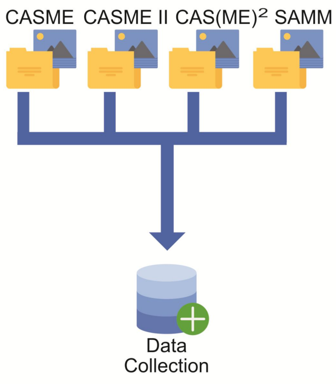


Figure 3: Data Collection

By using the excel files provided in each dataset, The Pandas library in Python was used to extract the location of *onset*, *apex*, and *offset* images in each subject directory. Each located image then categorized together into several emotion directories, such as *disgust*, *sadness*, *happiness*, *tense*, *repression*, and *others*. It first formed an imbalance dataset for each class. It can be shown in figure 4.

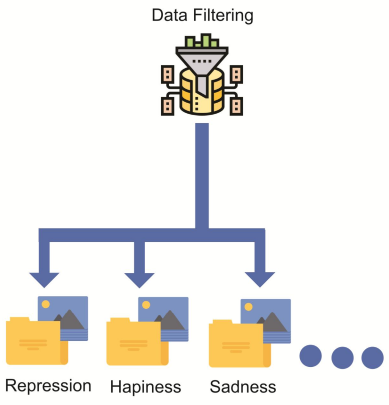


Figure 4: Data filtering into classes

The next step is to make it balance dataset, where each class was filtered to have maximum of 100 images. These images were from *onset*, *apex*, and offset frames only. To split into train and test dataset, the ratio of 75:25 being used, where train have 75 images and test dataset contained 25 images. This is according to equation 1, where *X* is the total of images in a class, *St* is the splitting threshold, and *Si* is the total images after split. The full formula is as shown in (1).

$$Si = X \text{ x } St \qquad (1)$$

Before splitting the images, the facial region on each image being cropped using Single Shot Detector (SSD) that used MobileNet as the baseline. Each image then being resized into 224 height and 224 width. After resizing without changing the colour channels, each image being saved in classes available in train, validation, and test dataset. This step is shown in figure 5.

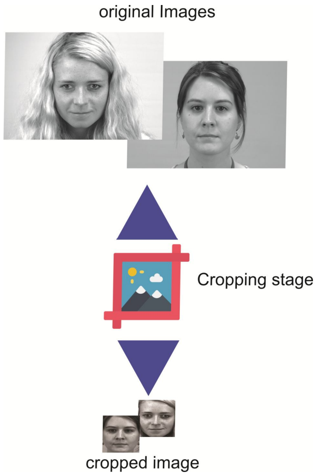


Figure 5: Image cropping step

This experiment used five models from *Residual Network* family, known as ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152. Another reason to use Residual Network is because of shortcut connection, which this connection could prevent vanishing gradient, cause by identity function as stated by [14].

This is because the weights initialized were the results after being trained for several datasets that proved to be robust. The number of models is known as *N*, while the number of repetition experiment is known as *E*, and the number of models produced after the experiments is *Mp*. If the number of experiments is 10, then in theory the total models produced are 40. This formula can write as in (2):

$$Mp = N \text{ x } E \qquad (2)$$

For the pre-trained models, these models were developed by using *Pytorch* library, a library exists in *Python* language. By using this library, these models do not have to reconstruct from beginning, instead these models and weights are provided together with this library. The model followed the original structure except at the end of fully-connected layers, where the output neurons only six, corresponded to the number of classes. The table 1 shows one of the of the model's structure for this experiment.

Table 1: Resnet18 structure by using Pytorch library

Layer name	Output	Structure
Conv1	112x112x64	7x7, 64, stride 2
Conv2_x	56x56x64	3x3 max pool, stride 2
		3x3, 64
		3x3, 64
Conv3_x	28x28x128	3x3, 128
		3x3, 128
Conv4_x	14x14x256	3x3, 256
		3x3, 256
Conv5_x	7x7x512	3x3, 512
		3x3, 512
Avg_pool	1x1x512	7x7, average pool
Fc	6	512x6 fully-connected
softmax	6	

These five models are using the same experimental settings, which the experiments being repeated 10 times, using 100 epochs, and use *Stochastic Gradient Descent* (SGD) as the optimizer. For the beginning, the learning rate being set as 0.001. Then, the same algorithms being tested with 0.0001 and 0.00001. These experiments were conducted by using NVIDIA GTX 1060 with memory size of 6GB. After trained the Convolutional Neural Network, each model in every experiment was recorded. Each model from 10 experiments are marginalized together to find the average. This goes the same as testing process, where each model being produced in each experiment was tested, thus the results were marginalized of being repeated 10 times.

4. RESULTS

In this section, the models being compared with each other in terms of performance by using different kinds of learning rates.

4.1 Performance of the ResNet models

There are three types of learning rates being used, which known as 0.001, 0.0001 and 0.00001. Each model was tested with these learning rates. Figure 8 shows the validation accuracy of each model for the learning rate of 0.001.

For the graph in figure 6, the *x-axis* is according to number of epochs, while *y-axis* is the accuracy that being normalized based on range of 0 to 1. As presented in the graph, ResNet101 and ResNet152 are the top models compared to ResNet18, ResNet34, and ResNet50. Despite ResNet152 became top model for learning rate 0.001, the accuracy dropped same as ResNet101. This indicated that having deeper architecture does not necessarily could achieve higher accuracy, given with limited data. This pattern can be observed through ResNet18 and ResNet50, where at the end of epoch, ResNet50 dropped right to the accuracy of ResNet18.

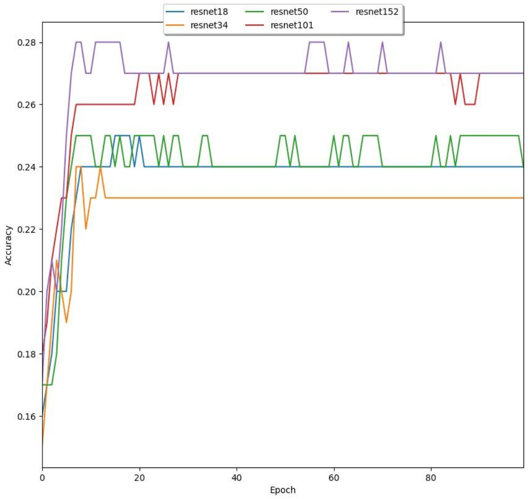


Figure 6: Performance of ResNet variants by using learning rate of 0.001

A graph in figure 7, Resnet18 dominated the best accuracy, same as ResNet101. However, compared to previous learning rate, the highest accuracy is a lower than the models trained on learning rate with 0.0001. Instead of gaining the best accuracy because of the depth of the network, ResNet152 is the lowest accuracy here. This might be because of identity shortcut and lack of data to be trained on. There are several theories that pointing on identity shortcut, where the models finally learn too little or no learning at all.

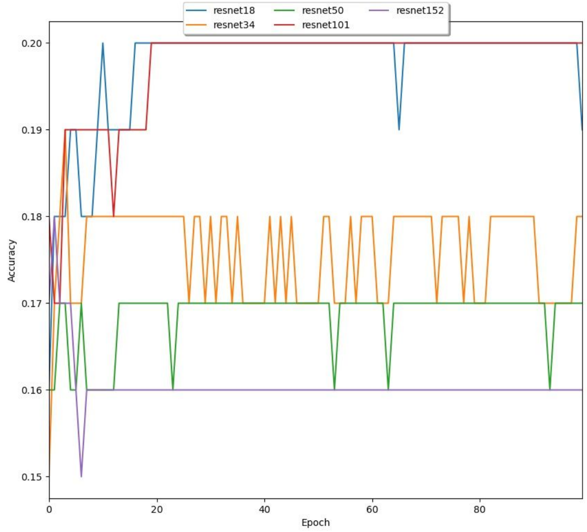


Figure 7: Performance of ResNet variants by using learning rate of 0.0001

In Figure 8, ResNet18 still achieved better accuracy, where ResNet152 achieved the lowest accuracy. The best accuracy here is below than any previous accuracy, and this can lead to a conclusion where it might be because of lower learning rate. From these graphs, it can be shown that smaller model tends not to break early, where amount of data to be used as training process is limited. Probably, if adding few of more epoch will show that the accuracy will slowly increase.

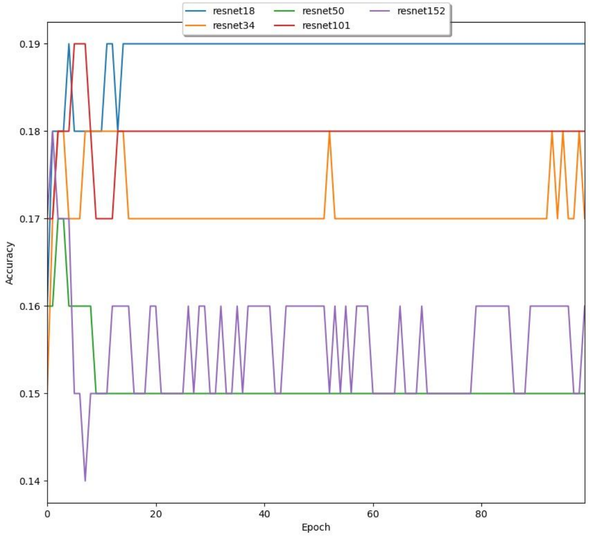


Figure 8: Performance of ResNet variants on learning rate 0.00001

After comparing each model trained by using different kinds of learning rates, it is safe to categorized ResNet18 is well performed models, despite by using limited amount of data. Therefore, the next section will further illustrate the performance of ResNet18, by comparing the performance of this model in a graph for both model's accuracies and losses.

4.2 Performance of the ResNet18

Figure 9 explicitly show the performance of ResNet18 closely in terms of training accuracies, when this model being tested with different kind of learning rate. As mentioned before, the best accuracy is depending on range of learning rates. By referring on this graph, learning rate of 0.001 easily dominated the highest accuracy can be achieved by pre-trained model. Despite of this, the model cannot achieve any better accuracy due to lack of training data.

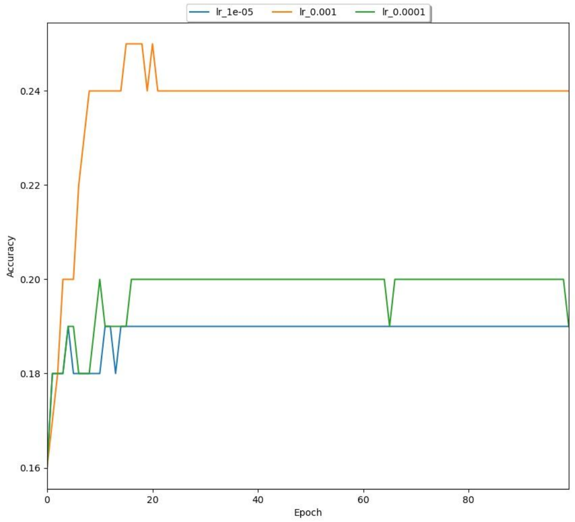


Figure 9: Performance of ResNet18 on different learning rates

Another important variable that need to take notes are the loss of training and validation phase. Figure 10 illustrated the loss performance for ResNet18. ResNet18 with 0.001 learning rate shows that it is overfitting. This might because of identity function, where previous data being added to current transformed information. Compared to learning rate of 0.0001, this model is under performed. This proves the theory of learning few information or not learning at all. ResNet18 with 0.00001 learning rate able to learn correctly. Despite the error during training phase has fluctuation, the error produced during validation phase is almost close enough with the error produced in training phase. The only drawback here is that this model might be improve from time to time, and it takes longer route to get better.

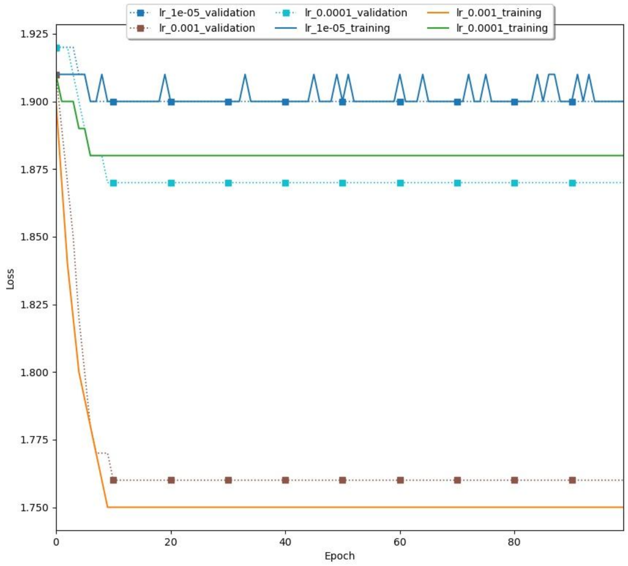


Figure. 10: Loss of training and validation for ResNet18

4.3 Discussion of the results

In this section, the overall experiments and performance on test datasets will be discussed. To recap the previous section, the best models for overall category with Stochastic Gradient Descent (SGD) is the ResNet18 with the learning rate of 0.00001 (1e-05).

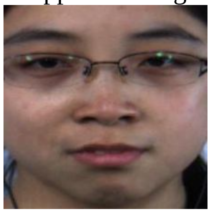
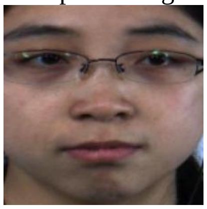
In table 2, the lowest accurate class is the happiness, with precision up to 8%, recall of 6% and f1-score with 6%. The highest accurate classification is the tense class, where precision is up to 18%, recall is 29% and f1-score is 16%. In table 7, for further analysis, the class of happiness and surprise will be compared, as the percentage between both classes are the closest in terms of f1-score, which surprise is up to 9%.

Table 2: Metrics on test data

	Precision	Recall	F1-Score
Disgust	0.1	0.18	0.12
Happiness	0.08	0.06	0.06
Other	0.12	0.17	0.13
Repression	0.14	0.21	0.14
Surprise	0.14	0.11	0.09
Tense	0.18	0.29	0.16
Accuracy	0.17	0.17	0.17

As shown in table 3, both images almost same as each other, where the left and right side of the mouth almost going upwards. Therefore, such classification error is making sense, since the models cannot differentiate both images, prompted to classify the wrong class.

Table 3: Comparison between happiness and surprise image from CASME datasets

Happiness Image	Surprise Image
	

Another note needs to be taken is that the validation accuracy is around 19% on the validation dataset. This result is not far from test dataset, where the models accurately predicted up to 17%. Moreover, the error rate of training and validation datasets are converged almost together. Thus, this model is the best because of the stabilization of training, validation, and testing accuracy.

***A.** Sample size*

For overall experiments, the sample size gathered from each database is smaller after going through data pre-processing stage. Thus, there are two ways of solving this problem. The first method is by making custom algorithms that can train using limited amount of data, or the second technique is by increasing the data itself, either by collecting a new data or by using augmentation of the existing data. Since the resources of collecting new data costing financial and time, augmentation is by far more preferred.

***B.** Imbalance data*

Another that needs to be addressed is the imbalance data. By right, each database has its own classes, which some of the classes have limited number of the data compared to another class provided in the datasets. The imbalance data influenced the sample size for training and testing datasets. If the models prepared to train by using the balance data, then the rule of thumb to prepare the data is by following the lowest amount of data available in a specific class. This however applied to datasets that do not use any kinds of augmentation technique.

***C.** Architecture*

The architecture of the models needs to be studied further. Many previous literature that related to the micro-expressions suggested to use lightweight models, which is true. This is one way to coup with a limited amount of data. However, it is good to see the current state-of-the-art performance, before incorporating either the contribution of these models or making new algorithm.

***D.** Epoch size*

Certainly, the epoch needs to be increased further, to find the actual pattern of the results. Stochastic Gradient Descent (SGD) is a quite robust optimizer, despite it often slower than any popular optimizers. One of the advantages available in SGD is that it is jumps to the nearest point, instead of randomly jumping around the gradients to find global minima. From here and now on, there will be several pre-trained models need to be tested, to find the benchmark of state-of-the-art models on a limited amount of data.

***E.** Optimizers*

Tons of optimizers that can be used, each of them can be matched up together with current state-of-the-art models. Therefore, it is suggested to use each optimizer on these models first and record each model's performance. Usually, SGD is most preferred because of its own robustness. The major drawback of SGD is that it is time consuming. This also related to epoch itself, where the epoch also needs to be longer than being presented on this paper.

***F.** Learning rates*

Learning rates is essential for models to learn. As presented in this paper, lowest learning rate could make the models to learn better but is not proportional to the architecture's depth. Probably, by making shallow network could work better with low learning rate. The future work that can be derived from this paper is to test shallow network via bigger and lower learning rates.

5. CONCLUSION

This paper presents the performance analysis of pre-trained Residual Neural Network on micro-expressions recognition. Micro-expressions are useful expressions for certain profession, such as psychiatrist or even law enforcer to detect deceits behavior. In this paper, five ResNet models, known as ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152 were trained on micro-expressions datasets by using pre-trained weights. Nevertheless, our results were not so encouraging and would demand few improvements as been suggested in the last section of this paper. In general, the performance of the models might not depend on the algorithmic structure itself. Instead the data also can be a major factor in the performance. Moreover, the micro-expressions itself happened not during specific timing, but it happened during certain milliseconds where it produced several related images and these images then become certain emotion as a whole.

This work used original data with limited samples without any augmentation. We intended to apply data augmentation and enhancement techniques, such as optical flow features which would improve the generalization of the model. In the future, this work will continue by using different kinds of existing models with its own pre-trained weights, thus more analysis can be done on the structure of the algorithm that may contribute the model performance.

ACKNOWLEDGEMENTS

We would like to appreciate the Universiti Teknologi MARA, supported with grant 600-IRMI/FRGS 5/3 (458/2019) under Ministry of Higher Education Malaysia. We appreciated the Chinese Academy of Science, for providing the micro-expressions databases, CASME I, CASME II, CAS(ME)^2 and to University of Manchester for providing us the SAMM databases.

REFERENCES

1. S. T. Liong and K. S. Wong, **Micro-expression recognition using apex frame with phase information**, *Proceedings - 9th Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2017*, vol. 2018-February. pp. 534–537, 2018, doi: 10.1109/APSIPA.2017.8282090.
2. X. Li *et al.*, **Towards reading hidden emotions: A comparative study of spontaneous micro-expression spotting and recognition methods**, *IEEE Trans. Affect. Comput.*, vol. 9, no. 4, pp. 563–577, 2018, doi: 10.1109/TAFFC.2017.2667642.
3. P. Ekman, ***Telling Lies: Clues to deceit in the marketplace, politics, and marriage***, New York: Norton, 1992, pp. 15-42.
4. L. F. Barrett, **Emotion Are Constructed**, in *How emotions are made: The secret life of the brain.* Boston, MA: Houghton Mifflin Harcourt, 2017, ch 2, pp. 25-41.
5. A. Freitas-Magalhães, C. Bluhm, M. Davis, and C. Alves, **Handbook on facial expression of emotion**, *Feel. Sci. Books*, no. November 2013, 2013, [Online]. Available: https://www.researchgate.net/publication/260796947_Handbook_on_Facial_Expression_of_Emotion.
6. O. I. Abiodun *et al.*, **Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition**, *IEEE Access*, vol. 7. pp. 158820–158846, 2019, doi: 10.1109/ACCESS.2019.2945545.
7. K. Fukushima, **Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position**, *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980, doi: 10.1007/BF00344251.
8. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, **Gradient-based learning applied to document recognition**, *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998, doi: 10.1109/5.726791.
9. A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, **A survey of the recent architectures of deep convolutional neural networks**, *Artif. Intell. Rev.*, 2020, doi: 10.1007/s10462-020-09825-6.
10. D. Choi, C. Shallue, Z. Nado, J. Lee, C. Maddison, and G. Dahl, **On Empirical Comparisons of Optimizers for Deep Learning**. 2019.
11. M. R. Shamsuddin, S. Abdul-Rahman, and A. Mohamed, **Shallow Network Performance in an Increasing Image Dimension**, *Communications in Computer and Information Science Soft Computing in Data Science*, vol. 652, pp. 3–12, 2016.
12. A. Krizhevsky, ***One weird trick for parallelizing convolutional neural networks***, 2014.
13. K. Simonyan and A. Zisserman, **Very deep convolutional networks for large-scale image recognition**, *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015. .
14. K. He, X. Zhang, S. Ren, and J. Sun, **Deep Residual Learning for Image Recognition**. 2015.
15. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. B. Wojna, ***Rethinking the Inception Architecture for Computer Vision***. 2016.
16. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, **Densely connected convolutional networks**, *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.
17. F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, **SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size**, pp. 1–13, 2016, [Online]. Available: http://arxiv.org/abs/1602.07360.
18. S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, **Aggregated residual transformations for deep neural networks**, *Proc. - 30th IEEE Conf. Comput. Vis. Pattern*

Recognition, CVPR 2017, vol. 2017-January, pp. 5987–5995, 2017, doi: 10.1109/CVPR.2017.634.

19. S. Zagoruyko and N. Komodakis, **Wide Residual Networks**, May 2016.
20. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, **MobileNetV2: Inverted Residuals and Linear Bottlenecks**, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.
21. M. Tan *et al.*, **Mnasnet: Platform-aware neural architecture search for mobile**, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 2815–2823, 2019, doi: 10.1109/CVPR.2019.00293.
22. N. Ma, X. Zhang, H. T. Zheng, and J. Sun, **Shufflenet V2: Practical guidelines for efficient cnn architecture design**, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11218 LNCS, pp. 122–138, 2018, doi: 10.1007/978-3-030-01264-9_8.
23. M. Aouayeb, W. Hamidouche, K. Kpalma, and A. Benazza-Benyahia, **A Spatiotemporal Deep Learning Solution for Automatic Micro-Expressions Recognition from Local Facial Regions**, *IEEE Int. Work. Mach. Learn. Signal Process. MLSP*, vol. 2019-October, pp. 2–7, 2019, doi: 10.1109/MLSP.2019.8918771.
24. Z. Xia, X. Hong, X. Gao, X. Feng, and G. Zhao, **Spatiotemporal Recurrent Convolutional Networks for Recognizing Spontaneous Micro-Expressions**, *IEEE Trans. Multimed.*, vol. 22, no. 3, pp. 626–640, 2020, doi: 10.1109/TMM.2019.2931351.
25. L. Zhou, Q. Mao, and L. Xue, **Dual-Inception Network for Cross-Database Micro-Expression Recognition**, in *2019 14th IEEE International Conference on Automatic Face Gesture Recognition (FG 2019)*, pp. 1–5, May 2019, doi: 10.1109/FG.2019.8756579.
26. L. Zhou, Q. Mao, and L. Xue, **Cross-database micro-expression recognition: A style aggregated and attention transfer approach,** *Proc. - 2019 IEEE Int. Conf. Multimed. Expo Work. ICMEW 2019*, pp. 102–107, 2019, doi: 10.1109/ICMEW.2019.00025.
27. Y. Liu, H. Du, L. Zheng, and T. Gedeon, **A Neural Micro-Expression Recognizer**, in *2019 14th IEEE International Conference on Automatic Face Gesture Recognition (FG 2019)*, pp. 1–4, May 2019, doi: 10.1109/FG.2019.8756583.