Neural Architecture Search Using Genetic Algorithm for Facial Expression Recognition

Shuchao Deng
College of Computer Science,
Sichuan University
Chengdu, China
dengshuchao@stu.scu.edu.cn

Yanan Sun*
College of Computer Science,
Sichuan University
Chengdu, China
ysun@scu.edu.cn

Edgar Galvan
Hamilton Institute, Dept. of CS,
Maynooth University, Naturally
Inspired Computation Res. Group
Ireland
edgar.galvan@mu.ie

ABSTRACT

Facial expression is one of the most powerful, natural, and universal signals for human beings to express emotional states and intentions. Thus, it is evident the importance of correct and innovative facial expression recognition (FER) approaches in Artificial Intelligence. The current common practice for FER is to correctly design convolutional neural networks' architectures (CNNs) using human expertise. However, finding a well-performing architecture is often a very tedious and error-prone process for deep learning researchers. Neural architecture search (NAS) is an area of growing interest as demonstrated by the large number of scientific works published in recent years thanks to the impressive results achieved in recent years. We propose a genetic algorithm approach that uses an ingenious encoding-decoding mechanism that allows to automatically evolve CNNs on FER tasks attaining high accuracy classification rates. The experimental results demonstrate that the proposed algorithm achieves the best-known results on the CK+ and FERG datasets as well as competitive results on the JAFFE dataset.

KEYWORDS

Facial expression recognition, network architecture search, genetic algorithm

1 INTRODUCTION

Facial expression recognition (FER) has a variety of applications in human society, such as medical care, automotive, and robotics manufacturing [4], to mention some. Convolutional neural networks (CNNs) are the most well-known networks thanks to their wide applicability in Euclidean data problems. These CNNs have become the standard architectures for FER tasks in multiple scientific works such as Deep-Emotion [15], thanks to outperforming other non CNNs techniques.

Architecture search is an area of growing interest as demonstrated by the large number of scientific works published in recent years and inspiring works have emerged for FER. For example, MnasNet-FER [1] builds a recurrent neural network controller that continuously adjusts the input architecture. Auto-FERNet [10] weakens the search space to a continuous spatial structure and then combines the greedy algorithm. The ConvGP [6] uses genetic programming through a series of crossovers and mutations. These neural architecture search algorithms using CNNs on FER tasks perform better than those hand-crafted CNNs. However, there are some limitations on these works such as requiring high computational power as well as making strong assumptions on the search space by using fixed length representations.

The main contribution of this work is addressing these issues. Specifically,

- We propose a variable-length encoding strategy to effectively address the need for a fixed-length encoding strategy.
- The skip connections are merged into the proposed algorithm to handle complex data.
- A global caching system is set up to reduce the computational cost of the evolution process.

2 RELATED WORK

2.1 Neural Architecture Search

Machine learning and deep learning are now being used in an increasing number of fields such as computer vision, healthcare, and robotics, thus requiring more and faster-automated design models. Google's proposal of NAS [21] caused a boom in the research community. Since then, NAS has attracted an increasing number of researchers due to its ability to automatically search for a good performing network [17]. There are three main parts in NAS: search space, search strategy, and performance evaluation.

The search space defines which architectures can be represented. A search strategy details how to explore and exploit the search space, which is often exponentially large or even unbounded. The goal of performance estimation is usually to find architectures that achieve high classification performance for unseen data.

2.2 Facial Expression Recognition

Many researchers have started to explore the combination of NAS and CNN. For example, Aghera et al. [1], proposed MnasNet-FER, an automatic mobile neural architecture-based approach for FER tasks. However, as pointed out by the authors, this approach is costly and difficult to balance between obtaining a lightweight architecture and obtaining a well-performing network. Li et al. [10], proposed Auto-FERNet. This performs an automatic search of neural architectures through gradient optimization with preset layers of network architecture and construction of hypernets, which often requires a great deal of expertise. Another noteworthy method is the one proposed by Evans et al. [6], dubbed ConvGP. This method overcomes the disadvantage of needing to pre-set the architecture length, but the excessive operators lead to requiring large computational calculations.

---

[*] Corresponding author.

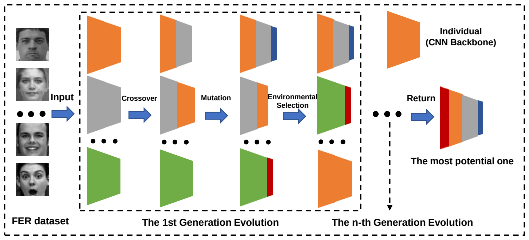

Figure 1: The general flow of the proposed algorithm.

For FER tasks, we use a genetic algorithm approach to automatically search for network architecture, which is due to its extraordinary results in numerous areas and we employ ingenious ideas to overcome the issues of NAS on FER.

3 METHODS

3.1 Algorithm Framework and Search Space

Figure 1 demonstrates the overall framework of the proposed algorithm. The FER dataset is used as input, and through a series of evolutionary generations, the most potential CNNs' architecture is automatically discovered. During the evolution process, firstly a population is randomly initialized to predefined population size, and the specific CNN architectures are encoded using the proposed encoding strategy. The fitness value of the individual is calculated, using accuracy as a metric. Then, the parent individuals are selected to generate new children according to the proposed crossover and mutation operators. Finally, the next generation of individuals is generated by selecting the parent and the new individuals, and the most potential one is returned.

The search space is defined by considering the following basic units: 3x3 *convolution*, 2x2 *maximum pooling*, 2x2 *mean pooling* and *skip connection*. The basic units form the basic blocks, where the skip connection layer consists of two convolutional layers and one skip connection. The proposed coding strategy is used to build the CNN architecture through these skip connections and pooling layers. The available numbers of feature maps are set to 64, 128, 256, 512 based on the settings employed by the state-of-the-art CNNs, and the step size is set to 1x1 (inspired by the ResNet series). The pooling layer is divided into mean pooling and maximum pooling, and the step size is also set to 1x1. Figure 2 shows an example of a CNN architecture implemented using our variable-length coding strategy.

3.2 Search Strategy and Performance Evaluation

The first step is to initialize the population. Each individual in the population encodes the network architecture. Next, we use the acceleration component to compute fitness values for the population shown in Algorithm 1. We set up a global cache system for each individual (Lines 1-4), and if we find that the model corresponding to the individual is already in the global cache system, then we can get its fitness value without training and validation (Lines 6-9), and vice versa, we record it (Lines 10-19). Finally, we return the population with their corresponding fitness values (Line 22).

Algorithm 1: Fitness evaluation with global cache

```
Input: The population $P_t$ of the individuals, the FER
       dataset, the training epochs, the training data $D_{train}$,
       the fitness evaluation data $D_{val}$;
Output: The population $P_t$ with individuals' fitness values;
1  if $t == 0$ then
2      $Cache \leftarrow \phi$ ;
3      Set $Cache$ to a global variable;
4  end
5  foreach individual in $P_t$ do
6      if the identifier of individual in Cache then
7          $v \leftarrow$ Query the fitness by identifier from Cache;
8          Set $v$ to individual;
9      else
10         $v_{best} \leftarrow 0$;
11         foreach epoch in the given training epochs do
12             Train the CNN on $D_{train}$;
13             $v \leftarrow$ Calculate the accuracy on $D_{val}$;
14             if $v > v_{best}$ then
15                 $v_{best} \leftarrow v$ ;
16             end
17         end
18         Set $v_{best}$ as the fitness of individual;
19         Put the identifier of individual and $v_{best}$ into Cache;
20     end
21 end
22 return $P_t$;
```

Algorithm 2 shows how crossover and mutation operators work. The first step in the algorithm is to select two parents using tournament selection of Size 2 (Lines 3-7). We designed single-point crossovers for variable-length coding for generating two offspring. (Lines 9-17). When mutating an individual, a specific mutation operation is selected from the provided mutation list (Lines 19-26). In the proposed algorithm, the available mutation operations defined in the mutation list are:

- adding a skip connection layer with random settings;
- adding a pooling layer with random settings;
- remove the layer at the selected location; and
- randomly changing the parameter values in the selected location building block.

The first two mutation operators increase the network depth and the third mutation operator decreases the network depth. Crossover and mutation followed by return of offspring (Line 27).

Finally, there is an environmental selection to form the next generation of individuals, and we use a binary tournament selection, as mentioned before, as well as elitism.

4 EXPERIMENTS

4.1 Benchmark Datasets

The Extended Cohn-Kanade (CK+) dataset [11] contains 593 video sequences from a total of 123 different subjects, ranging from 18 to 50 years of age with a variety of genders and heritage. We divided

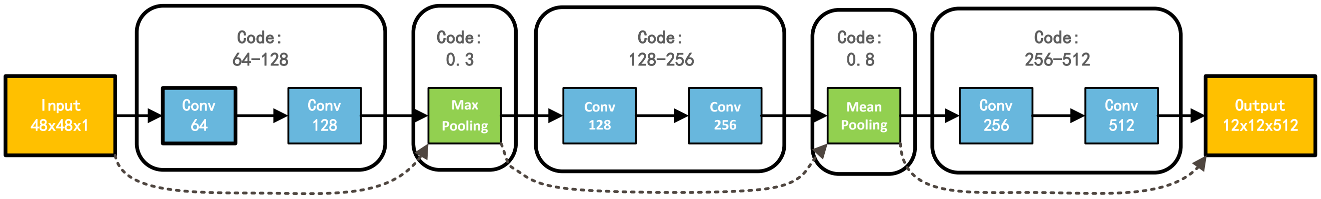

Figure 2: A network architecture formed with the proposed coding strategy.

Algorithm 2: Offspring generation

```
Input: The population $P_t$ containing individuals with
       fitness, the probability for crossover operation $p_c$,
       the probability for mutation operation $p_m$, the
       mutation operation list $l_m$ , the probabilities of
       selecting different mutation operations $p_l$ ;
Output: The offspring population $Q_t$;
1  $Q_t \leftarrow \phi$ ;
2  while $|Q_t| < |P_t|$ do
3      $p_1 \leftarrow$ Randomly select two individuals from $P_t$, and
         from the two then select the better one;
4      $p_2 \leftarrow$ Repeat Line 3;
5      while $p_2 == p_1$ do
6          Repeat Line 4;
7      end
8      $r \leftarrow$ Randomly generate a number from (0,1);
9      if $r < p_c$ then
10         Divide $p_1$ into two random parts by a point;
11         Divide $p_2$ into two random parts by a point;
12         $o_1 \leftarrow$ The first part of $p_1$ adds the second part of $p_2$;
13         $o_2 \leftarrow$ The first part of $p_2$ adds the second part of $p_1$;
14         $Q_t \leftarrow Q_t \cup o_1 \cup o_2$;
15     else
16         $Q_t \leftarrow Q_t \cup p_1 \cup p_2$;
17     end
18 end
19 foreach individual p do
20     $r \leftarrow$ Randomly generate a number from (0,1) ;
21     if $r < p_m$ then
22         $i \leftarrow$ Randomly choose a point in $p$;
23         $m \leftarrow$ Select one operation from $l_m$ in $p_l$;
24         Do the mutation $m$ at the point $i$ of $p$;
25     end
26 end
27 Return $Q_t$;
```

Table 1: Comparison of accuracy with peer competitors.

Method	Pre-train	CK+	JAFFE	FERG	Manual or Auto
Deep-Emotion [15]	No	98.00	92.80	99.30	Manual
Ensemble Multi-feature [20]	No	-	80	97	Manual
Adversarial NN [7]	Yes		-	98.2	Manual
LBP [9]	No	93.9	88.3	96.7	Manual
SAFER [19]	No	96.37	95.30	-	Manual
FAN [14]	Yes	99.69	-	-	Manual
AFER [18]	No	-	96.05	-	Manual
FERIK [5]	Yes	97.59	-	-	Manual
HMTL [16]	Yes	98.23	79.88	-	Manual
ViT [3]	Yes	98.17	94.83	-	Manual
ViT + SE [3]	Yes	99.80	92.92	-	Manual
ViT [3]	No	98.57	88.23	-	Manual
ViT + SE [3]	No	99.49	90.61	-	Manual
Auto-FERNet [10]	No	98.89	**97.14**	-	Auto
ConvGP [6]	No	-	96.67	-	Auto
Ours	No	**100**	95.71	**99.98**	Auto

the dataset into training set, validation set and test set, the numbers of which are 687, 101 and 193 respectively. The JAFFE dataset [12, 13] consists of 213 images of different facial expressions from 10 different Japanese female subjects. We used 120 images for training, 23 images for validation, and 70 images for the test (10 images per emotion in the test set). FERG [2] is a database of cartoon characters with annotated facial expressions containing 55,769 annotated face images of six characters. We use around 34k images for training, 14k for validation, and 7k for testing.

4.2 Parameter Settings

We set the population size to 20 and the evolutionary generations to 20. The crossover and mutation probabilities in the parameter settings for this algorithm to search for the most potential CNN architecture are set to 0.9 and 0.2. We trained a total of 600 epochs using stochastic gradient descent with an initial learning rate of 0.025, a momentum of 0.9, and a weight decay of 3e-4 and adjusted the learning rate to 0.017 at 100 epochs, 0.001 at 300 epochs, and 0.0001 at 500 epochs. In addition, the number of available feature maps is set to 64,128, 256, 512 according to the settings used in state-of-the-art CNN. For the FERG dataset, we set the epochs to 20 and keep the rest of the values the same. The algorithm can be stopped when the evolutionary generation is over or the performance is good. The results of all experiments were averaged over five tests.

4.3 Experimental Results

To show the effectiveness and efficiency of the proposed algorithm, we selected the state-of-the-art algorithms as peer competitors.

As shown in Table 1, we achieved 100% accuracy on the CK+ dataset and 99.98% accuracy on the FERG dataset, both achieving the best-known results so far, and 95.71% accuracy on the JAFFE dataset, higher than most models, achieving competitive results. Furthermore, compared to manual design, our algorithm is completely free of manual design. In particular, in contrast to the currently existing methods that perform well only on a single dataset,

Table 2: The number of layers and parameters of the network architecture and the time spent in the architecture search.

Dataset	Layers	Parameters (M)	Time (GPUhs)
CK+	16	10.2	29
JAFFE	12	5.9	27.75
FERG	13	11.07	16

our proposed algorithm performs well on multiple datasets, thus demonstrating the good adaptability of our proposed algorithm.

4.4 Analysis of Results

There are three key aspects that these modern and revolutionary methods have focused their attention on: size of the network, the number of parameters and the time to train these networks. These three elements are shown in Table 2.

As it can be seen, the number of layers varies for each of the datasets used in this work. This shows how the proposed mutation operators work effectively to automatically adjust the size of the network, with 16, 12 and 13 layers for the CK+, JAFFE and FERG datasets, respectively. The number of parameters is also reported in this table, third column from left to right. Finally, we show how it is possible to effectively train our network without requiring massive computational power. In this work, we use an NVIDIA 2080 Ti GPU card. We can see that the time, measured in GPU-hours, goes from 16 to +27 hours, for the CK and the other two datasets, respectively. These times show how our proposed encoding is effective on FER tasks, where it has been well documented that multiple high-end GPUs are required to train these types of networks within reasonable time [8].

5 CONCLUSION AND FUTURE WORK

We propose an algorithm for automatically designing network architectures based on genetic algorithms for FER. Specifically, we propose a variable-length encoding strategy and the corresponding crossover operator to efficiently explore the optimal network depth. Secondly, skip connections are introduced into the algorithm to make it possible to handle complex data. Finally, a global caching system is set up to reduce the computational cost of the evolution process. Experimental results show that our algorithm achieves good performance. In the future, we will work on accelerated fitness evaluation methods to apply the proposed algorithms to larger datasets.

REFERENCES

[1] Saumya Aghera, Hariyali Gajera, and Suman K Mitra. 2020. MnasNet Based Lightweight CNN for Facial Expression Recognition. In *2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*. IEEE, IEEE, Gunupur Odisha, India, 1–6.

[2] Deepali Aneja, Alex Colburn, Gary Faigin, Linda Shapiro, and Barbara Mones. 2017. Modeling stylized character expressions via deep learning. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II 13*. Springer, 136–153.

[3] Mouath Aouayeb, Wassim Hamidouche, Catherine Soladie, Kidiyo Kpalma, and Renaud Seguier. 2021. Learning Vision Transformer with Squeeze and Excitation for Facial Expression Recognition. , 13 pages. http://arxiv.org/pdf/2107.03107v4

[4] Kévin Bailly, Séverine Dubuisson, et al. 2017. Dynamic pose-robust facial expression recognition by multi-view pairwise conditional random forests. *IEEE Transactions on Affective Computing* 10, 2 (2017), 167–181.

[5] Zijun Cui, Tengfei Song, Yuru Wang, and Qiang Ji. 2020. Knowledge augmented deep neural networks for joint facial expression and action unit recognition. *Advances in Neural Information Processing Systems* 33 (2020), 12.

[6] Benjamin Evans, Harith Al-Sahaf, Bing Xue, and Mengjie Zhang. 2018. Evolutionary deep learning: A genetic programming approach to image classification. In *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, IEEE, http://ieeexplore.ieee.org/servlet/opac?punumber=8466244, 1–6.

[7] Clément Feutry, Pablo Piantanida, Yoshua Bengio, and Pierre Duhamel. 2018. Learning Anonymized Representations with Adversarial Neural Networks. arXiv:1802.09386 [stat.ML]

[8] Edgar Galván and Peter Mooney. 2021. Neuroevolution in deep neural networks: Current trends and future challenges. *IEEE Transactions on Artificial Intelligence* 2, 6 (2021), 476–493.

[9] Durga Ganga Rao Kola and Srinivas Kumar Samayamantula. 2021. A novel approach for facial expression recognition using local binary pattern with adaptive window. *Multimedia Tools and Applications* 80, 2 (2021), 2243–2262.

[10] Shiqian Li, Wei Li, Shiping Wen, Kaibo Shi, Yin Yang, Pan Zhou, and Tingwen Huang. 2021. Auto-FERNet: A Facial Expression Recognition Network with Architecture Search. *IEEE Transactions on Network Science and Engineering* 8, 3 (2021), 10.

[11] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. 2010. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-workshops*. IEEE, 94–101.

[12] Michael J Lyons. 2021. " Excavating AI" Re-excavated: Debunking a Fallacious Account of the JAFFE Dataset. *arXiv preprint arXiv:2107.13998* (2021).

[13] Michael J Lyons, Miyuki Kamachi, and Jiro Gyoba. 2020. Coding facial expressions with Gabor wavelets (IVC special issue). *arXiv preprint arXiv:2009.05938* (2020).

[14] Debin Meng, Xiaojiang Peng, Kai Wang, and Yu Qiao. 2019. Frame attention networks for facial expression recognition in videos. In *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, IEEE, Taipei, Taiwan, 3866–3870.

[15] Shervin Minaee, Mehdi Minaei, and Amirali Abdolrashidi. 2021. Deep-emotion: Facial expression recognition using attentional convolutional network. *Sensors* 21, 9 (2021), 3046.

[16] Mahdi Pourmirzaei, Farzaneh Esmaili, and Gholam Ali Montazer. 2021. Using Self-Supervised Co-Training to Improve Facial Representation. *CoRR* abs/2105.06421 (2021), 15. arXiv:2105.06421 https://arxiv.org/abs/2105.06421

[17] Yanan Sun, Bing Xue, Mengjie Zhang, and Gary G Yen. 2019. Evolving deep convolutional neural networks for image classification. *IEEE Transactions on Evolutionary Computation* 24, 2 (2019), 394–407.

[18] Yacine Yaddaden, Mehdi Adda, and Abdenour Bouzouane. 2021. Facial Expression Recognition using Locally Linear Embedding with LBP and HOG Descriptors. In *2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH)*. IEEE, IEEE, Boumerdes, Algeria, 221–226.

[19] Yacine Yaddaden, Mehdi Adda, Abdenour Bouzouane, Sebastien Gaboury, and Bruno Bouchard. 2018. User action and facial expression recognition for error detection system in an ambient assisted environment. *Expert Systems with Applications* 112 (2018), 173–189.

[20] Hang Zhao, Qing Liu, and Yun Yang. 2018. Transfer learning with ensemble of multiple feature representations. In *2018 IEEE 16th International Conference on Software Engineering Research, Management and Applications (SERA)*. IEEE, IEEE, Kunming, China, 54–61.

[21] Barret Zoph and Quoc V. Le. 2016. Neural Architecture Search with Reinforcement Learning. *CoRR* abs/1611.01578 (2016), 16. arXiv:1611.01578 http://arxiv.org/abs/1611.01578