University JEAN MONNET

TER

2022/2023

Facial Expression Classification using Machine Learning

Authors:
Bao Khang DO PHAN

June 2023

Summary

In recent years, manga has become a very popular subject around the world. Inspired by its potential in various domains like games and animation, many studies have attempted to classify automatically facial expressions of manga characters.

The results of my 3-months-research project using neural network models such as VGG-16, ResNet-50, GoogLeNet represent a positive sight of the use of Machine Learning in image classification for manga universe. The highest accuracy achieved in this study is 51% by a ResNet-50 model, relatively higher than existing models on the same subject. Even though not as accurate as the traditional approach k-NN with 76% accuracy, machine learning methods still have more room for improvement and great potential to be the best with further studies on the use of data-set and the networks.

Contents

1 Introduction

1.1 Context

Recently, there has been more and more needs to categorize automatically manga (Japanese graphic novels) images in order to enrich storytelling, provide valuable insights, as well as opening up new creative possibilities in various domains where manga culture and artistry are present.
There has been already a few existing studies on this subject such as:

- Classification using PyTorch with ResNet-50 by MERT KÖKLÜ[1]. The highest accuracy reached by this experiment is 42% on Figure1.

```
Training:
Train Epoch: 1 [0/300 (0%)]     Loss: 1.307388
Train Epoch: 1 [75/300 (25%)]   Loss: 1.987805
Train Epoch: 1 [150/300 (50%)]  Loss: 2.105261
Train Epoch: 1 [225/300 (75%)]  Loss: 1.290375

 Testing:
Test set: Average loss: 49.1651, Accuracy: 26/150 (17%)


Training:
Train Epoch: 2 [0/300 (0%)]     Loss: 1.261794
Train Epoch: 2 [75/300 (25%)]   Loss: 1.404785
Train Epoch: 2 [150/300 (50%)]  Loss: 1.291204
Train Epoch: 2 [225/300 (75%)]  Loss: 1.131131

 Testing:
Test set: Average loss: 1.9434, Accuracy: 52/150 (35%)


Training:
Train Epoch: 3 [0/300 (0%)]     Loss: 1.052421
Train Epoch: 3 [75/300 (25%)]   Loss: 0.763120
Train Epoch: 3 [150/300 (50%)]  Loss: 1.070751
Train Epoch: 3 [225/300 (75%)]  Loss: 0.854838

 Testing:
Test set: Average loss: 1.7767, Accuracy: 63/150 (42%)
```

Figure 1: The result of ResNet-50 using PyTorch

- Application of DenseNet201 for classification by STPETE ISHII[2].
  The research shows the highest accuracy achieved by this experiment is 32% as shown as below on Figure 2.

	precision	recall	f1-score	support
0	0.17	0.17	0.17	6
1	0.60	0.23	0.33	13
2	0.22	0.33	0.27	6
3	0.13	0.25	0.17	8
4	0.33	0.30	0.32	10
5	0.67	0.33	0.44	6
6	0.56	0.71	0.63	7
accuracy			0.32	56
macro avg	0.38	0.33	0.33	56
weighted avg	0.40	0.32	0.33	56

Figure 2: Table of the confusion matrix

Figure 3: Visualization of the accuracy and loss of the model

- Other existing methods: Transfer learning by STPETE ISHII[3]; Application of Hitesh Kumar's Emotion Detection using DCNN[4]; Application of DCNN for classification[5].

The current models have a range of accuracy of 30 to 40%. In an attempt to understand and improve if possible this accuracy, I conducted a 3-months research on image classification working with machine learning models such as VGG-16[6], ResNet-50[7] and GoogLeNet[8].

1.2 Problem

The main goal of this study is to evaluate the potential of neural network models and machine learning algorithms in classifying the facial expressions of manga characters.

1. This research focused mainly in the first step on implementing existing machine learning models and algorithms, for example VGG-16 or GoogLeNet.

2. The second step is to improve if possible their performance with parameter tuning and Transfer Learning.

3. And lastly, the resulted models will be compare with other methods in order to position the efficiency of the models in the research with existing state-of-art models as well as a more traditional approach like k-Nearest Neighbor (k-NN).

As a result, I achieved a highest accuracy around 50% for ResNet-50, and around 20-30% for VGG-16 as well as GoogLeNet.
These results are rather coherent compare to existing models of other researches but still lower compare to the results of k-NN (around 76%). The detailed results will be presented later in the report.

The machine learning methods are proved to have potential in facial expression classification but will need more researches to reach higher accuracy.

2 Materials and Methods

2.1 Materials

2.1.1 Data-set

The data-set was created and published on Kaggle by MERT KÖKLÜ [9]. It contains 463 photos of manga characters faces with various expressions:

1. Sad (57 photos)
2. Pleased/Satisfied (38 photos)
3. Shock/Surprised (103 photos)
4. Angry (55 photos)
5. Crying (56 photos)
6. Happy (87 photos)
7. Embarrassed (67 photos)

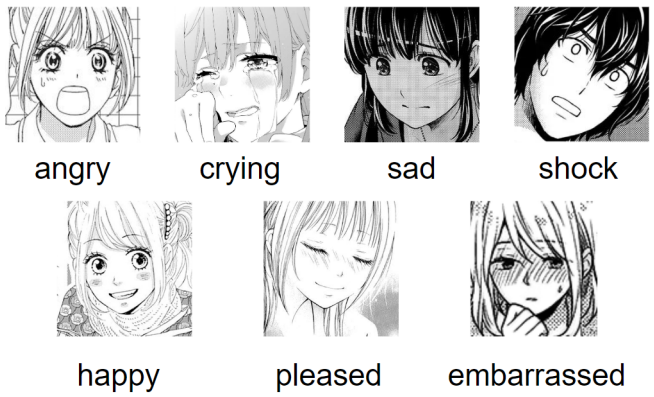

Figure 4: Example images of each category

2.1.2 Neural networks

The models will be represented in this research are:

1. VGG-16[6]: a variants of VGGNet, a convolutional neural network model that have 16 layers.
2. ResNet-50[7]: a 50-layer convolutional neural network.
3. GoogLeNet[8]: a 22-layer deep convolutional neural network that's a variant of the Inception Network, a Deep Convolutional Neural Network developed by researchers at Google.

All the model used in this research are CNN (Convolutional Neural Network), one of the Machine Learning models. They allow you to build highly accurate and intelligent systems. Therefore, CNN has many applications, especially for problems that need to identify objects in images (Object Recognition).

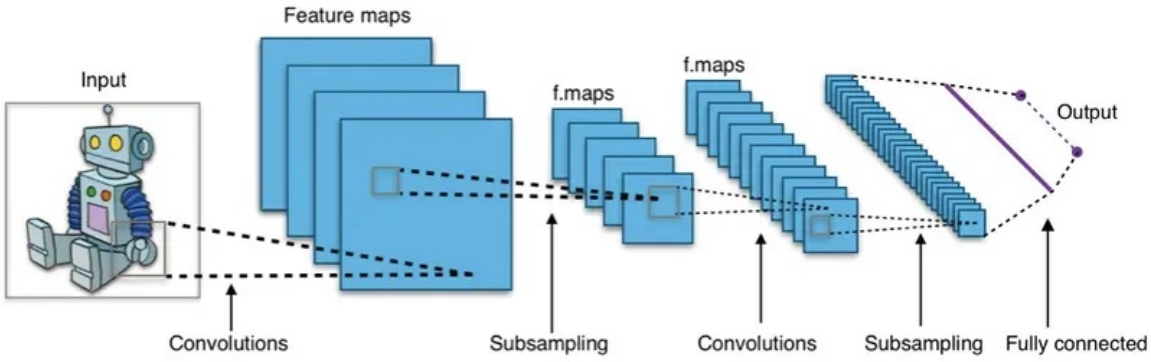

Figure 5: CNN architecture

2.2 Methods

2.2.1 Data Preparation

Firstly, the data-set is divided randomly into 2 sets: train (80%, approximately 370 photos) in order to create the models and test (20%, approximately 90 photos) used for evaluating.
The images must then be processed as follow before feeding into the neural network:

1. Converting images to grayscale.

2. Resizing the images: Each photo will have a final dimension of 100x100.

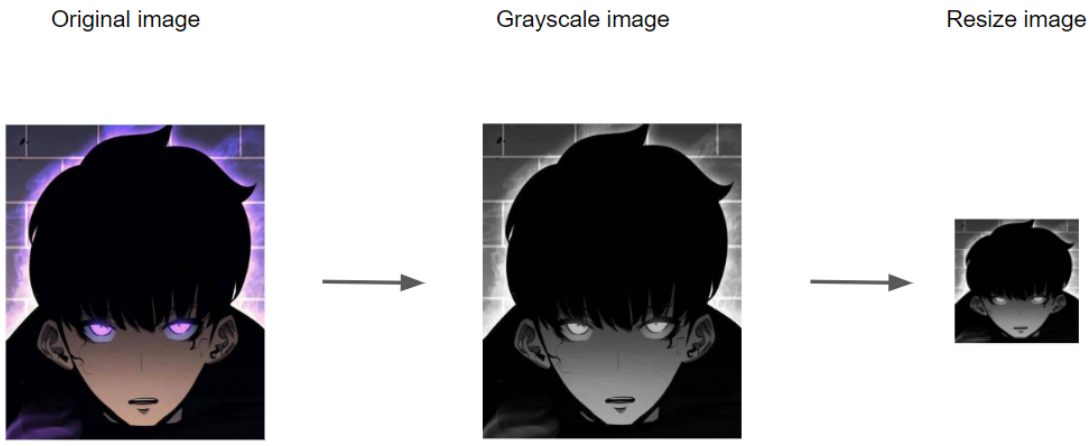

Figure 6: Processing image

In the next steps, we will be working with pixels matrix instead of the images themselves.

3. Converting the images into pixel values.

4. Normalizing pixel values.

5. Labeling the images.

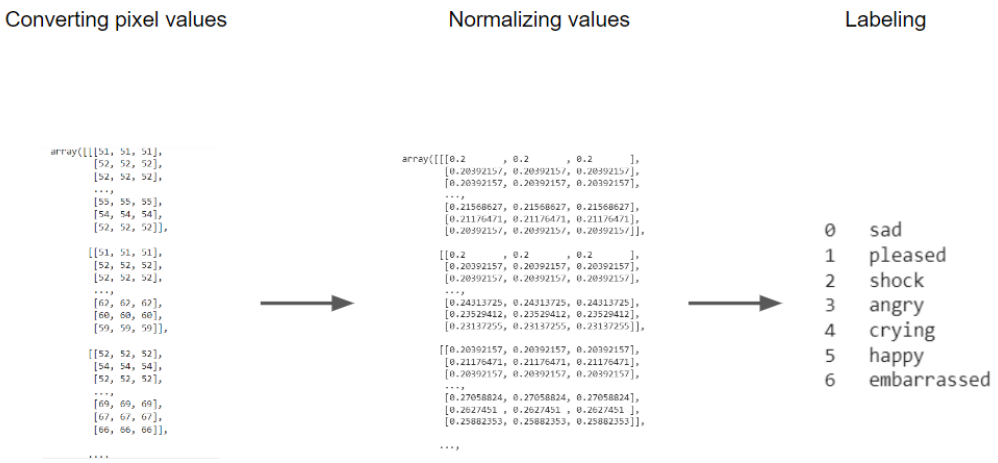

Figure 7: Processing pixels matrix and labeling

In the end, for each picture, we have a matrix that representing the images pixel by pixel and its labels.

2.2.2 Model Training

When a model processes an image, the input data goes through layers. At each layer, feature maps representing the occurrence of a particular pattern or feature, are created by applying a convolution filter to the input data.

During training, it is important to have hyper-parameter tuning in order to help the models to be more effective and accurate. Hyper-parameters are fixed parameters that are set before the learning process such as learning rate, number of hidden layers and neurons, number of epochs,...
In this research, the two parameters we will be focused on are :

1. Learning rate: the rate helps in smoothing the learning process of the model during training and accelerating convergence at the same time. Starting at the learning rate equals to 0.001, we then monitor training progress and validation evaluation. If the model converges too quickly or too slowly, we adjust the learning rate using Learning Rate Finder[10]. With this tool, we are able to visualize the changes of the loss over the learning rate range and from that pick out the optimal learning rate to use for the model.

2. Number of epochs: mostly to avoid over-fitting. At first we train the models with 30 epochs to see how them performs in general, then monitor the lowest point on the loss or use Early Stopping[11] to avoid over-fitting.

   We will also looking quickly at other parameters such as :
   - **Training time** : the time a model takes to complete a training process which includes train on a training set and test on a validation set.
   - **CPU or Central Processing Unit consumption**: CPU is the primary component of a computer that performs most of the processing tasks and calculations required during machine learning training. The metric shows which model is the most costly in term of CPU.

2.2.3 Model Evaluating

1. Evaluation metrics :
   - **Accuracy**: measure the overall correctness of the model. It is calculated by dividing the correct predictions made by the model over all the samples in the data-set. A high accuracy indicates that the model is making accurate predictions, while a low accuracy suggests that the model is making more errors.
   However, accuracy alone may not provide a complete picture of the

model's performance. There are other metrics like precision, recall, F1 score that can be used with it to get a more nuanced evaluation.

- **Loss**: show how well the model processed during training. The choice of loss function depends on the type of problem being addressed, for example with our problem, we use the categorical cross-entropy for multi-class classification.

In order to know if a model is well trained, meaning not over-fitting nor under-fitting, it is critical to consider its accuracy and loss:

- **Over-fitting**: The problem occurs when the model is too fitted with the training data set, including noises of said data-set, so it can not perform well with other data. In the context of accuracy and loss: the training accuracy is very high but so is the loss on both the validation and test set.

- **Under-fitting**: The problem occurs when the model is too generalized but not as accurate. As a result, its performance is poor on both training and testing time because the model fails to minimize the loss effectively while training and performs poorly on new unseen data while testing. In the context of accuracy and loss, the accuracy is low and the loss is high on both the train and the test data.

2. Activation map

   An activation map[12] is a representation of the feature maps. It represents the positions and degrees of the features detected by a particular filter. It highlights the regions of the input image that are most significant for that particular feature by a lighter version of that color. We can also use heat map to clarify the activation map with the cold tone color for less significant and hot tone color for more significant.

Activation maps are important tools for understanding and explaining the behavior of CNN and their ability to extract relevant features from input data.

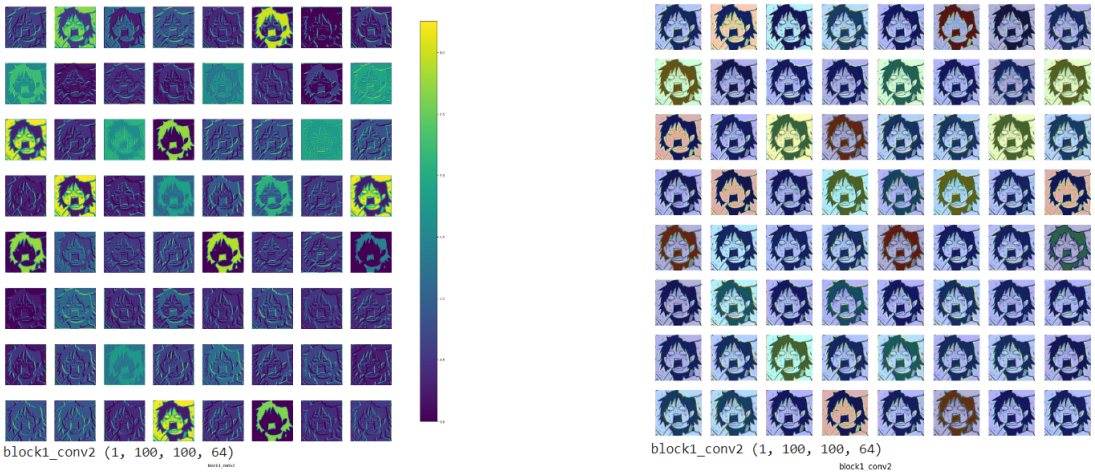

Figure 8: Activation map and Heat map of one of the layer

2.2.4 Model Enhancing using Transfer Learning

Transfer learning is a technique where knowledge gained from training a model on one task or data-set is utilized to improve the performance of the same model on a different but related task or data-set. In transfer learning, a pre-trained model, which has already learned useful feature representations from a large data-set, is re-purposed for a new task or data-set, often with limited labeled data.

2.2.5 Comparison with other method: k-NN

In this experiment, we also use k-NN to classify images and compare its accuracy with the machine learning models.

3 Results

3.1 Model Training

1. Hyper parameter tuning

    - Learning rate:

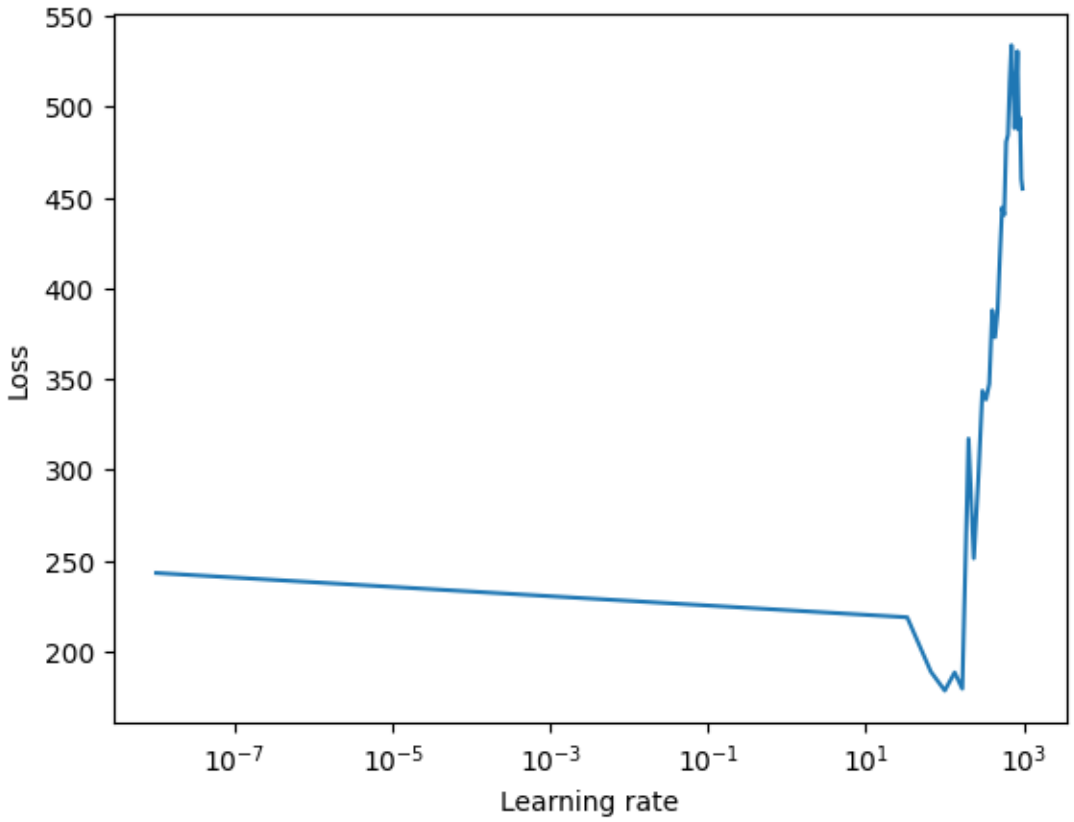

Figure 9: The changes of loss according to the learning rate

As shown on Figure 9, we can observe that the loss go through 3 phases according to the values of the learning rate :

- From the lowest learning rate to a rate of 10^1: No significant change, the loss function does not improve.
- From there to a learning rate near 10^2: The loss function drops down and descends to some minimum value.
- From a rate of 10^2 to 10^3: The loss function ascends very fast and starts to diverge.

As a result, the best range of the learning rate is between 10^1 and 10^2 where the loss function reaches its minimum.

- Number of epochs:

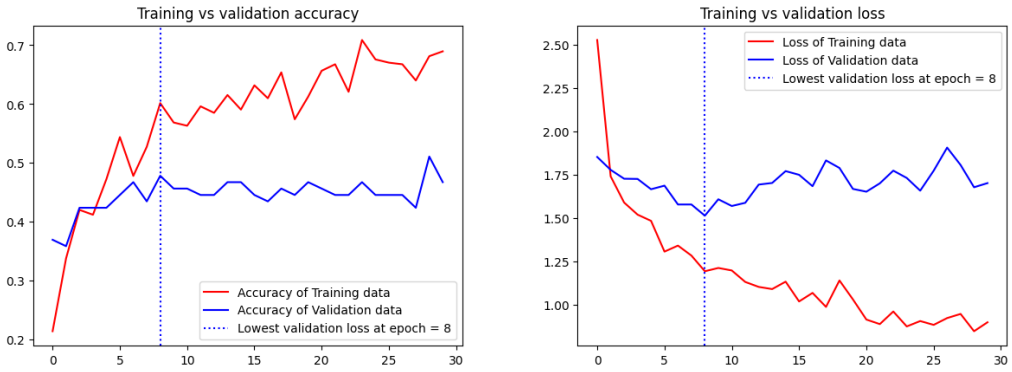

Figure 10: Tuning with 30 epochs

While processing with 30 epochs on Figure 10, we can easily monitor the lowest point of the loss on the validation set. The same result can be achieved with Early Stopping tool, as shown below on Figure 11. The optimal number of epochs observed is 8.

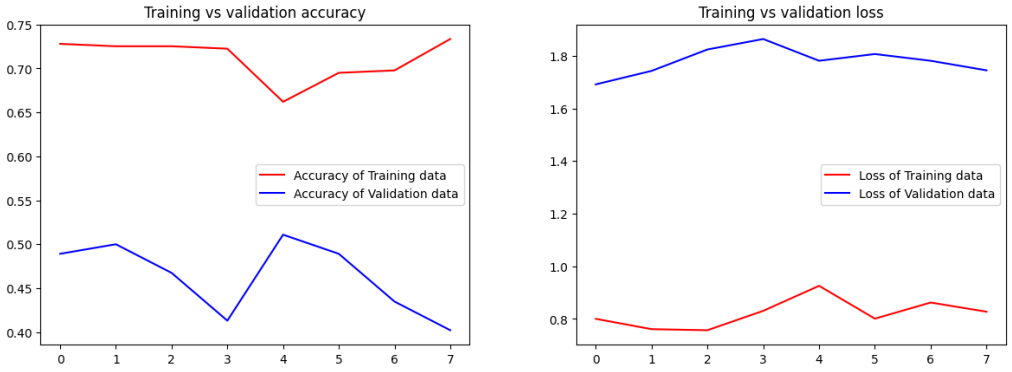

Figure 11: Tuning with Early Stopping

2. Training time

   As shown as below on Figure 12: ResNet-50 has relatively lower training time comparing to other models.

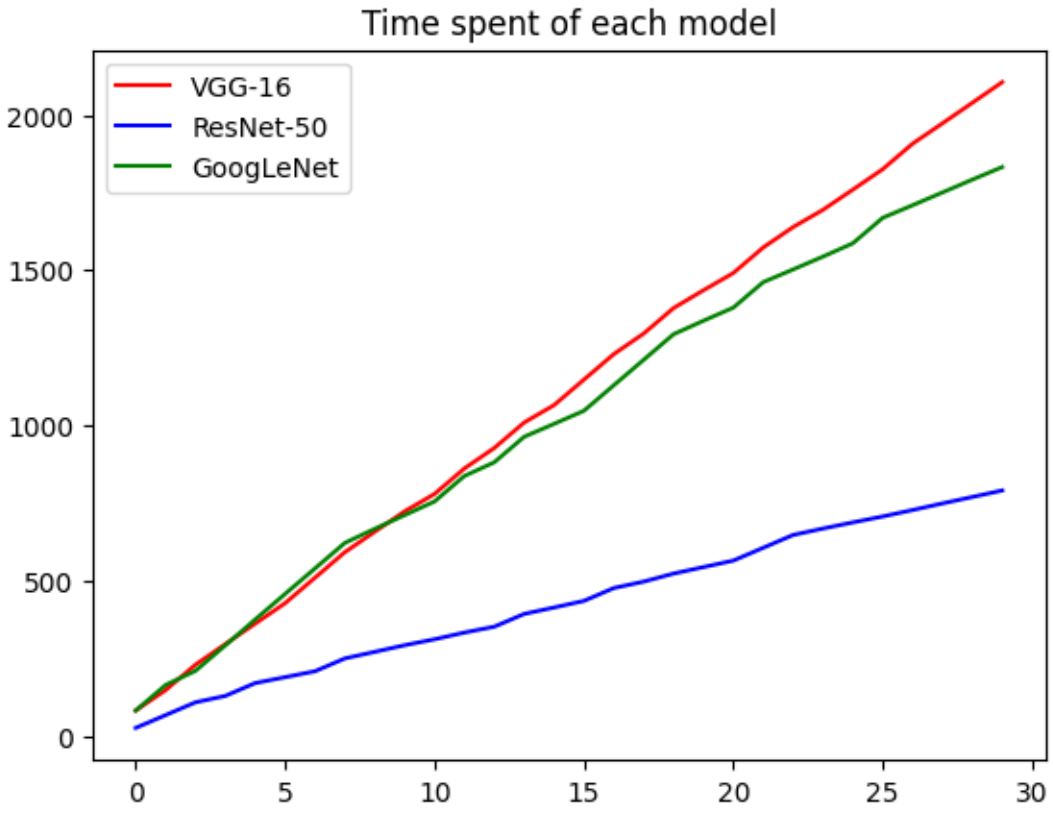

Figure 12: Representation of the time performance of the 3 models

3. CPU consumption

   As observed on Figure 13, GoogleNet consumes the least CPU while ResNet-50 consumes the most.

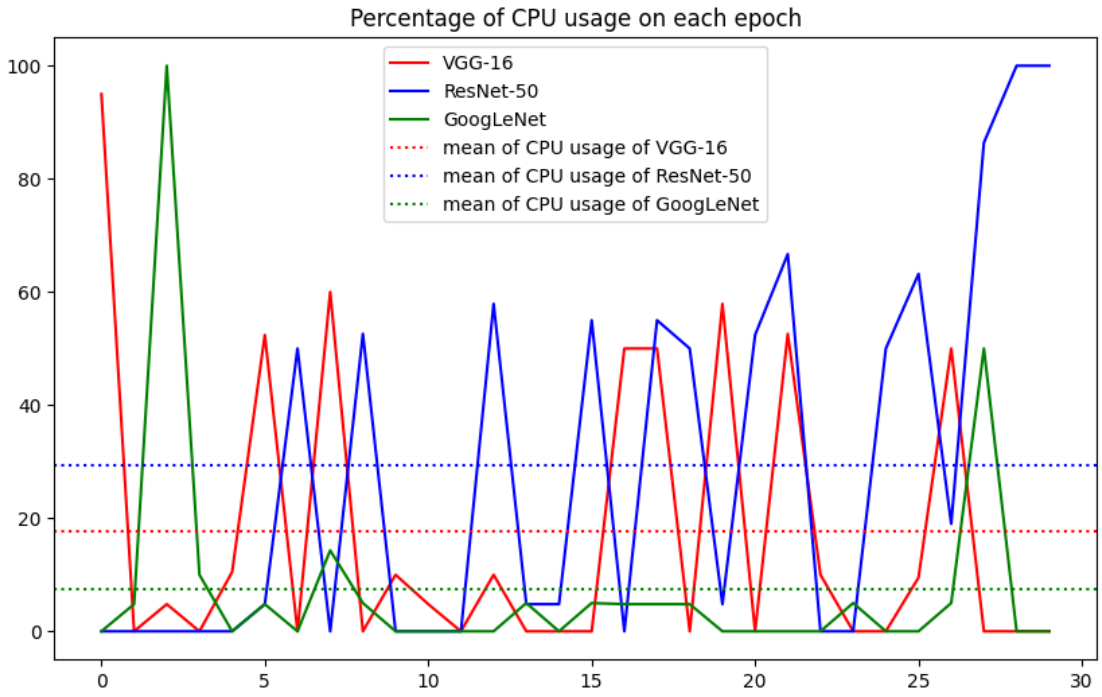

Figure 13: Representation of CPU cost of the 3 models

3.2 Model Evaluating

- ResNet-50 : The resulted model can classify well all classes with an overall accuracy of 51%. Class 2 (shock) has the best results among the 7 classes.

	precision	recall	f1-score	support
0	0.50	0.30	0.37	10
1	0.33	0.14	0.20	7
2	0.65	0.72	0.68	18
3	0.36	0.42	0.38	12
4	0.17	0.09	0.12	11
5	0.55	0.86	0.67	21
6	0.60	0.46	0.52	13
accuracy			0.51	92
macro avg	0.45	0.43	0.42	92
weighted avg	0.48	0.51	0.48	92

Figure 14: Detailed evaluation of ResNet-50 model

- VGG-16 : Among the 7 facial expressions, the resulted model can only recognize the class 2 (shock) and 5 (happy). It has an overall low accuracy of 28% with class 5 being categorized slightly better than class 2.

```
3/3 [==============================] - 12s 3s/step
```

	precision	recall	f1-score	support
0	0.00	0.00	0.00	10
1	0.00	0.00	0.00	7
2	0.28	0.44	0.34	18
3	0.00	0.00	0.00	12
4	0.00	0.00	0.00	11
5	0.30	0.86	0.44	21
6	0.00	0.00	0.00	13
accuracy			0.28	92
macro avg	0.08	0.19	0.11	92
weighted avg	0.12	0.28	0.17	92

Figure 15: Detailed evaluation of VGG-16 model

- GoogLeNet : The model can only recognize the class 2 (shock) and has an overall 20% accuracy.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	10
1	0.00	0.00	0.00	7
2	0.20	1.00	0.33	18
3	0.00	0.00	0.00	12
4	0.00	0.00	0.00	11
5	0.00	0.00	0.00	21
6	0.00	0.00	0.00	13
racy			0.20	92
avg	0.03	0.14	0.05	92
avg	0.04	0.20	0.06	92

Figure 16: Detailed evaluation of GoogLeNet model

3.3 Model Enhancing with Transfer Learning

At first, the base model (ResNet-50) has around 50% accuracy at validation. The loss is decreasing well at training time and stable at validation time.

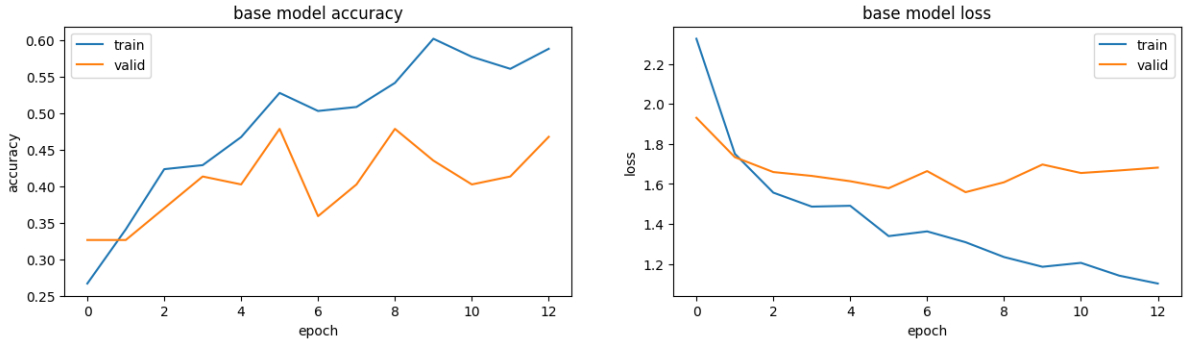

Figure 17: Evolution of base model accuracy and loss

After implementing the Transfer Learning method, the model performs well at training time but very poor at validation.

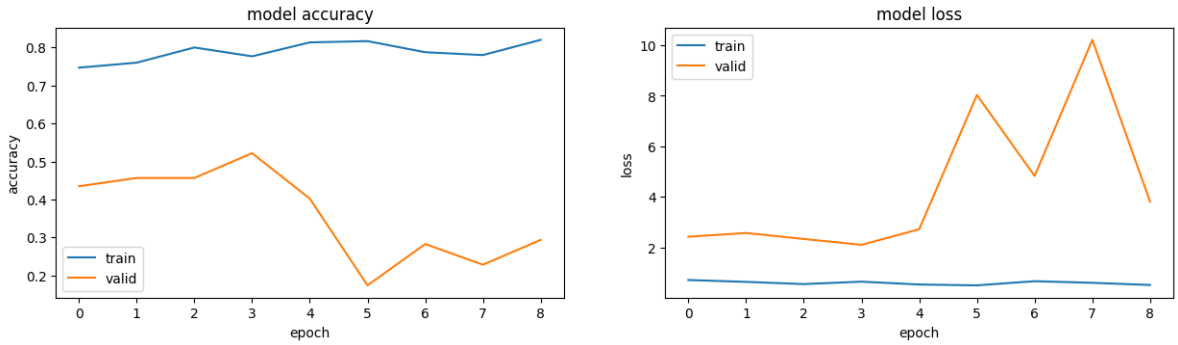

Figure 18: Evolution of the model accuracy and loss after Transfer Learning

Compare to the model after Transfer Learning, it is clear that the base model is better in terms of accuracy and loss.

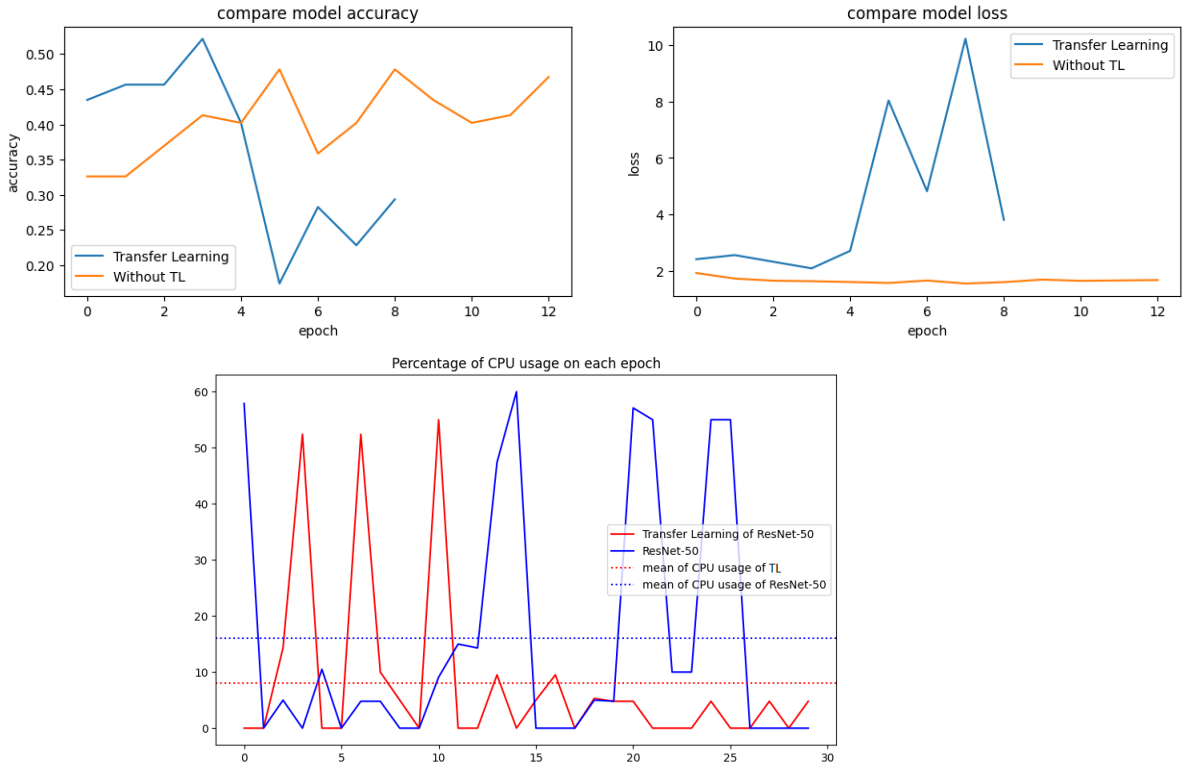

Figure 19: Comparison of the models before and after Transfer Learning

3.4 K-NN Accuracy

The algorithm reached 76% accuracy at the best k=19.

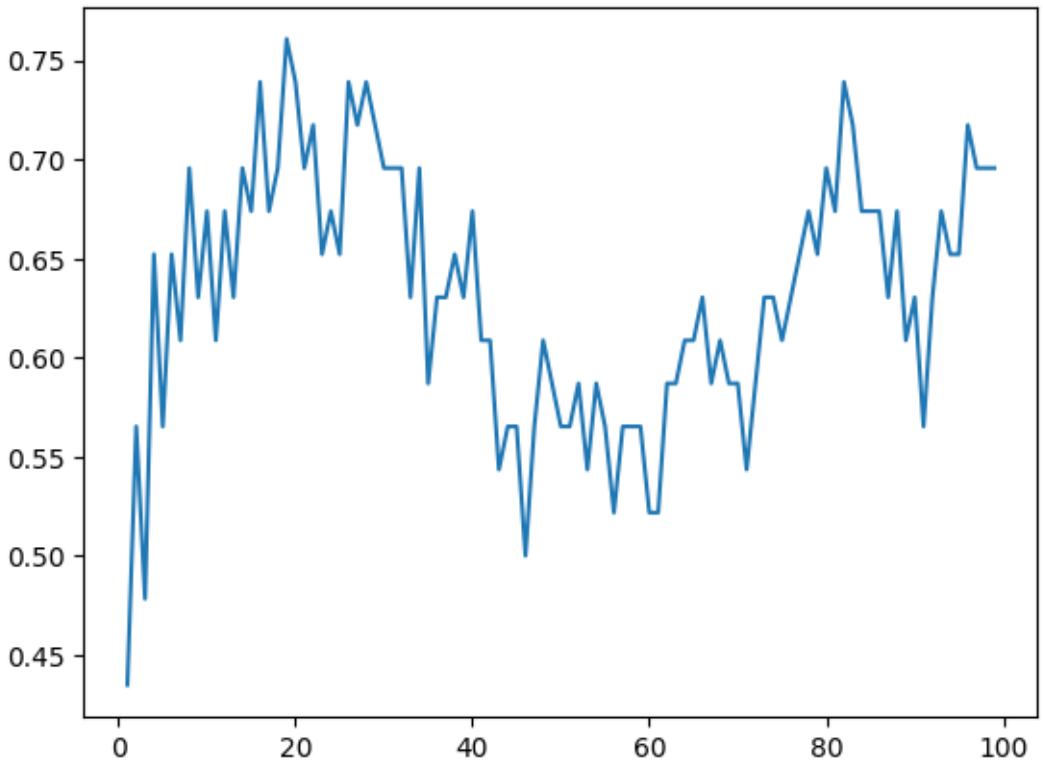

Figure 20: K-NN performance

4 Discussion

4.1 Model Training

1. Time processing: Despite having the most layers, ResNet-50 took the least time to finish computing.

2. CPU consumption: Probably because ResNet-50 has the most layers to go through (50 layers), it consumes on average the biggest percentage of CPU.
   Strangely in case of the other 2 models, VGG-16 with fewer layers than GoogLeNet (16<22) costs more in term of both training time and CPU consumption. That probably is because of the depth of each layer and number of parameters of the model, VGG-16 with more parameters (nearly 15 millions parameters in total) is costlier than GoogLeNet (approximately 6 millions parameters in total).

```
=================================================================
Total params: 14,781,255
Trainable params: 66,567
Non-trainable params: 14,714,688
_________________________________________________________________
```

```
=================================================================
Total params: 6,392,645
Trainable params: 6,392,645
Non-trainable params: 0
_________________________________________________________________
```

Figure 21: Detailed number of parameters of VGG-16 and GoogLeNet

4.2 Model Evaluating

ResNet-50 clearly outperformed VGG-16 and GoogLeNet in many aspects:

1. Recognition of different classes: ResNet-50 can classify all 7 classes while the others 2 can only manage 1 or 2. Especially in the case of the class "shock", it is well classified by all the models with a very high precision. The fact that the characters have their mouth and eyes wide open in most images of this class probably makes it easier for the models to recognize and distinguish them from other classes.

Figure 22: Examples images of the class "shock"

Activation and Heat map can also be used to verify this hypothesis.

Figure 23: Sample from the Activation and Heat maps

The hot tone color such as red representing areas that are more focused on by the models is found mostly in the bottom half of the image where the mouth of the character is usually located.

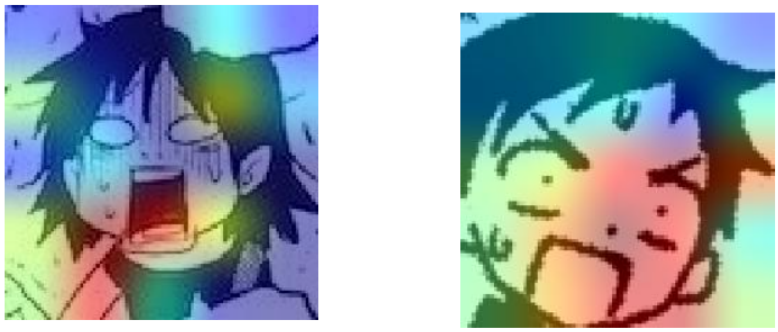

Figure 24: Heat images from STPETE_ISHII's research [3]

2. Accuracy: Since ResNet-50 is able of classify more classes, it also has higher accuracy.
   Moreover if we look closely in the details: precision and recall, it turns out that ResNet-50 performs better even in the same class as VGG-16 and GoogLeNet.

In comparison with resulted models in other researches, for example DenseNet201 with the accuracy of 32%, the accuracy of VGG-16 and GoogleNet are around the same range.
ResNet-50 however has a much higher accuracy, at around 50%. Based on the obtained results, ResNet-50 is the most suited model to classify our data-set.

4.3 Model Enhancing with Transfer Learning

Unfortunately, the model ResNet-50 after conducted with the Transfer Learning method shows no improvement compare to the base model. It could probably be that the pre-trained model with Transfer Learning is over-fitting because the accuracy and loss are super high at training time but validation set got a really low accuracy.
As a result the only thing the method improved is saving more CPU because it can exploit the knowledge gained from being pre-trained.

4.4 Comparison with other method: k-NN

Compare to the machine learning models, k-NN algorithm has the best result on the data set with a 76% accuracy. However, traditional approaches like k-NN are more difficult to find room for improvement while machine learning methods can still have great potential to be improved.

5 Conclusion

In conclusion, through this research, we are able to create a ResNet-50 based model with a higher accuracy compare to the results of other researches (50% $>$ 30%). However, when compare to a traditional method such as k-NN, our best model still has lower accuracy at the moment (50% $<$ 76%).
These results show that the Neural Network models have the potential in image classification but will need better understanding of the networks and data set. With the right hyper-parameters and a good use of the data set, we might achieve higher and higher accuracy that could surpass the traditional methods in further studies.

6 References

[1] Classification using PyTorch with ResNet-50 - MERT KÖKLÜ (2021)
https://www.kaggle.com/code/mertkkl/pytorch-manga-facial-expression-classification

[2] Application of DenseNet201 for classify - STPETE ISHII (2021)
https://www.kaggle.com/code/stpeteishii/manga-face-densenet201

[3] Transfer learning - STPETE ISHII (2021)
https://www.kaggle.com/code/stpeteishii/transfer-learning-for-manga-facial-expressions

[4] Application of Hitesh Kumar's Emotion Detection using DCNN - MARÍLIA PRATA (2021)
https://www.kaggle.com/code/mpwolke/manga-facial-expressions

[5] Application of DCNN for classification - VISHAL KALATHIL (April 2023)
https://www.kaggle.com/code/vishalkalathil/manga-facial-expression-classifier-dcnn

[6] API of VGG-16 model
https://keras.io/api/applications/vgg/#vgg16-function

[7] API of ResNet-50 model
https://keras.io/api/applications/resnet/#resnet50v2-function

[8] Implementation of GoogLeNet model - KHUYEN LE (March 2021)
https://medium.com/mlearning-ai/implementation-of-googlenet-on-keras-d9873aeed83c

[9] Manga facial expression data-set - MERT KÖKLÜ (2021)
https://www.kaggle.com/datasets/mertkkl/manga-facial-expressions

[10] Learning Rate Finder method - SIMONA MAGGIO (June 2020)

https://blog.dataiku.com/the-learning-rate-finder-technique-how-reliable-is-it

Implementation of Learning Rate Finder - JEREMY JORDAN (March 2018)
https://www.jeremyjordan.me/nn-learning-rate/

[11] API of Early Stopping
https://keras.io/api/callbacks/early_stopping/

[12] Activation map - TIRTHAJYOTI SARKAR (September 2019)
https://towardsdatascience.com/activation-maps-for-deep-learning-models-in-a-few-lines-of-code-ed9ced1e8d21