# Pre-build Convolutional Neural Networks Application On Facial Expression Classification

Sinh Thanh Nguyen
25099704

Ngoc Quang Vinh Pham
25100660

Cong Tuan Minh Le
25165123

## Introduction

In the age of human-machine interaction and automated systems, it is more important than ever to comprehend human emotions. Emotion Classification, a subfield of Artificial Intelligence and Machine Learning, seeks to close this gap by teaching computational models to recognize and interpret human emotions. This report describes emotion classification models that use facial expression data to distinguish seven distinct emotions: Surprise, Anger, Happiness, Sadness, Neutral, Disgust, and Fear.

### 1. Purpose of the Project

This project's primary objective is to create and evaluate pre-built Convolutional Neural Network architectures that can classify human emotions based on facial expressions. The utility of such a model is both extensive and significant, with applications ranging from customer experience enhancement to mental health monitoring.

### 2. Data Source

The dataset utilized for this project was derived from the "Challenges in Representation Learning: Facial Expression Recognition Challenge (1)." The dataset consists of 28,079 training samples and 7,178 testing samples and is comprehensive and well-structured. Each sample is a 48x48 pixel grayscale image with automatically registered faces to ensure a uniform layout.

### 3. Methodology Summary

Our method included data pre-processing, models selection, and models evaluation at various stages. These steps are described in greater detail in the following sections of this report.

### 4. Participation

This project aims to contribute to the expanding field of emotion recognition by developing an empirically validated classification model for facial expressions. The project also evaluates the performance of various Convolutional Neural Network architectures to offer insights into their effectiveness for emotion classification tasks.

In this report, we will delve deeper into the methodologies employed, discuss the results obtained, and provide suggestions for future research in this exciting field of study.

## Problem Statement

In the context of the ongoing research project, we are faced with a complex challenge in the field of machine learning. Since the aim of this project is to develop a robust and accurate model capable of classifying facial expressions accurately, it becomes essential to list the main issues that need to be resolved as we move into the next phases of this project. This problem statement serves as the essential basis of our report.

To tackle the difficulties presented by the facial emotions classification problem and set the platform for further study, we suggest the following solutions:

(1) *Data Augmentation:* Data augmentation techniques will be crucial because of the inherent complexity of facial expressions and the potential of class imbalance. These techniques involve generating additional training samples by applying transformations like rotation, scaling, and cropping to diversify the dataset (2). The purpose of data augmentation is to improve the model's ability to detect minor variations in expressions and to balance the depiction of various emotions.

(2) *Model Selection:* The success of this project depends critically on selecting the appropriate model architecture. Model selection is the procedure of selecting one final classification model for a training dataset from a pool of candidate CNN architectures (3). Models that are adept at capturing facial characteristics and subtle emotional nuances are necessary due to the unique nature of facial expressions.

(3) *Hyperparameter tuning:* To maximize the model's performance, hyperparameters must be fine-tuned after a model architecture has been chosen (3). This entails modifying dropout rates, batch sizes, learning rates, and other architecture-specific parameters. The iterative process of hyperparameter tuning aims to maximize the model's capacity to reliably identify and categorize emotions.

Through the data augmentation implementation, careful model selection, and comprehensive hyperparameter tuning, our anticipated outcomes encompass the following: improved models' performance, improved capacity to capture subtle emotional cues and variations in facial features, optimization of hyperparameters resulting in a highly accurate and robust facial emotions classification model.

This problem statement concludes by outlining our methodical approach to solving the unique obstacles presented by the facial emotional classification task. The effective implementation of data augmentation, model selection, and hyperparameter tuning has the potential to substantially progress machine learning research, especially in facial expression identification. The results of this study could have a significant impact on technology related to human-computer interaction and emotion-aware AI systems.

## Literature Review

First and foremost, let us consider the concept of Convolutional Neural Network (CNN). CNNs are types of deep learning model which are designed to analyze grid-like data which make them fundamental for image classification problems. A digital image is a representation of graphical data in binary form. It comprises a series of pixels arranged in a grid, along with pixel values indicating the brightness and color of each pixel. With the given data, the CNNs models detect patterns or features through spatial hierarchies, enabling the network to learn increasingly complex information, as well as pooling layers to reduce dimensionality and computational complexity, while preserving essential features. In computer vision tasks such as image classification, image segmentation, and object detection, these networks have obtained significant results. CNNs could autonomously acquire complicated features, eliminating the need for laborious feature engineering.

In addition, CNNs are highly adaptable to various input quantities and capable of handling complex patterns and data variations, however for each structure of the machine learning models, the input shape can be variated to optimize the efficiency of the prediction.

To make our project more efficient and realistic, we have researched several CNN architectures to find the advantages and disadvantages of the "Net", the input requirement to optimize its proficiency, and evaluate both classic and modern structures to classify which is the best fit for our problem.

Let's look at the architecture GoogLeNet network (1), also called the Inception architecture which is a Deep Convolutional Neural Network constructed by researchers at Google. It's indisputable fact that GoogLeNet is a good architecture, but it has some vital computational demands that may make it impractical for certain real-time or resource-constrained applications.

When the GoogLeNet based on a stacked inception approach the VGG16 is modelled based on the notion that depth is essential for performance precision and visual representations. It's constructed from 16 weight layers and assess the networks with very small (3x3) convolution filter to generates about 138 million parameters with the input tensor size of VGG16 is (224, 224) with 3 RGB channel. As of my last knowledge update in September 2021, VGG16 was indeed regarded as one of the finest computer vision models.

Nevertheless, several new architectures and model variants have been developed since then, some of which outperform VGG16 in a variety of computer vision tasks. Typically, these models have greater depth, more sophisticated architectural features, and enhanced training methods. One of those are the ResNet architecture (Residual Networks) that developed residual connections, which significantly enhanced the training of extremely deep neural networks. ResNet50 is designed as a bottleneck to improve training performance by addressing the level of saturation degradation issue with residual mapping. It is demonstrated that this architecture has the greatest performance in terms of region of focus, however, the worst in robustness accuracy when facing unseen data form different sources (4). However, we still need some adjustments when applying these architectures into the project to get the objective sight of how they performed with our given dataset.

AlexNet is a deep convolutional neural network architecture that has acquired popularity in computer vision tasks, especially image classification. When AlexNet first was introduced, it demonstrated impressive results in image object recognition. It gained popularity because it was more complex (had more layers) and made use of clever techniques to increase accuracy (5).

An alternative is MobileNetV1 which can be applied in term of a compact and more effective architecture. MobileNet is a streamlined architecture that makes use of depth-wise separable convolutions to build lightweight deep convolutional neural networks and offers an effective model for this project. A previous work [reference] posted on Kaggle using MobileNet shows that:

- Evaluate Public Test Accuracy: 44.92%
- Evaluate Private Test Accuracy: 45.42%

This show that the model's performance is moderate but may not be very accurate. However, the plot that show the model accuracy and loss shows quite promising results when the prediction of the model is stable and not fluctuate so we can relatively anticipate the differences between the accuracy score of the training and test set *(Appendix 1).* Depending on the specific task and the difficulty of the dataset, an accuracy score in the mid-40s indicates that the model's performance is moderate but may not be very accurate. If higher precision is required, it is essential to consider alternative evaluation metrics and potentially investigate methods to enhance the model's effectiveness. Additionally, it is essential to comprehend the task's context and the characteristics of the datasets to interpret these scores accurately. One of the things that we can observe is the imbalance of the input dataset when the number of 'disgust' attributes is fundamentally lower than the rest, something that we can address in our project.

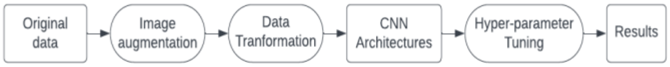
Here is the table show the design comparison of all the Architectures above:

| Architecture | Depth | Parameters | Layers | Input Size |
|---|---|---|---|---|
| VGG16 | 16 | 138 M | 41 | 224x224 |
| GoogLeNet | 22 | 7 M | 144 | 224x224 |
| ResNet50 | 50 | 25.6 M | 177 | 224x224 |
| AlexNet | 8 | 60 M | 24 | 227x227 |
| MobileNetV1 | 28 | 4.2 M | 88 | 224x224 |

*(Table 1)*

## Approach

As mentioned, several pre-built Convolution Neural Networks would be applied into the data. However, all of them would be processed as below:



Firstly, we would perform pre-processing and overall analysis on the original raw data to obtain some of its general understandings. Then, to gain more input data and reduce the imbalance between categories, Image Augmentation would be applied.

Next, the data would be split into train and validation sets, used for measuring training loss and accuracy, and test set, used to conclude models' performance by different metrics, such as accuracy score, f1-score, AUC score and confusion matrix. However, each pre-built Convolution Neural Network architecture requires a different input image format, for instance, GoogleNet is compatible with images size 224x224, same as VGG16, whereas 227x227 is suitable for AlexNet. Therefore, we would perform Input Transformation to convert input images data into favorable sizes beforehand.

After gathering those records, we could then decide which architecture the most effectively performs, moreover, applying hyper-parameter tuning to improve its efficiency.

Because of hardware limitation, cloud computing platform Kaggle would be utilized throughout the project. Kaggle offers with several GPU selections, which reduce processing time on complex pre-built Neural Network architectures.

## Data

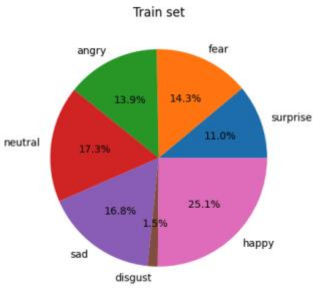
### 1. Data Acquisition:

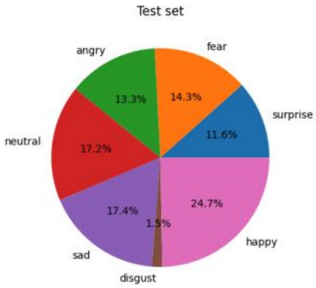
The dataset FER-2013 was imported from Kaggle. It consists of grayscale facial images sizing 48x48 and faces are more of less centered and occupy the same amount of space in each image.

Also, images are already registered with labels (0 = Angry, 1 = Disgust, 2 = Fear, 3 = Happy, 4 = Sad, 5 = Surprise and 6 = Neutral). The dataset has yet been divided into train and test sets, including 28708 and 3589 examples respectively.

### 2. Data Overview

Images in train and test set based on their labels are distributed respectively as follow:





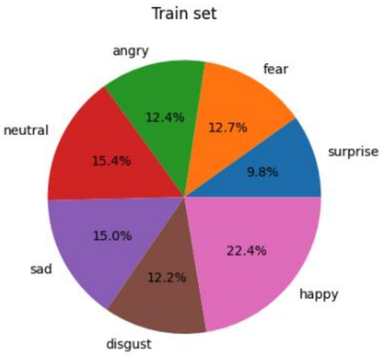
'Happy' occupied both train and test sets at most, around 25%, while each other categories covered from 10% to 20%. However, the figure also clearly shows that number of 'disgust' images only occupied 1.5% of both datasets.

Because of the distributions, the data set is considered imbalanced.

### 3. Data manipulation

To reduce the data set imbalance, Image Augmentation was applied on 'disgust' set of the training data. The augmentation techniques consist of horizontal flip and 45-degree rotation, and the target is to increase the number of 'disgust' images to approximately 3500, which is relatively equivalent to other categories'. This process would be done by ImageDataGenerator from Keras - TensorFlow. After that, the augmented images would be added to the original train data.

After image augmentation, the data distributed better as follow:



As previously mentioned, all images in the dataset are with size of 48x48 and they would have to be transformed into architectures' favorable sizes, which are shown in *(Table 1)*.

After being transformed, the data set was normalized and split into train, validation, and test set. The test set has already initially been defined with 3589 examples. The train and validation data mutually share the same source, which is the original training data, with the percentages of 80% and 20% respectively.

## Conclusion

### 1. Summary

The report above partially describes our work of looking for the most effective pre-trained Convolutional Neural Network architecture into facial expression problem. It has briefly summarized the process of exploratory data analysis and data preparation before being flushed into the 4 architectures.

### 2. Future work

The raw data has already been improved and ready for the next stage of our project: Model selection and development, where we would dive dipper into architectures' framework, along with their performance on the data, and architectures improvement by hyper-parameter tuning techniques.

# References

[1] *Challenges in Representation Learning: Facial Expression Recognition Challenge | Kaggle*. (n.d.). https://www.kaggle.com/competitions/challenges-in-representation-learning-facial-expression-recognition-challenge/data

[2] Shorten, C. and Khoshgoftaar, T.M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1). doi:https://doi.org/10.1186/s40537-019-0197-0

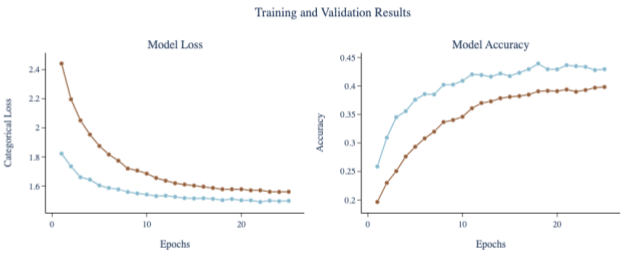
[3] Brownlee, J. (2019). A Gentle Introduction to Model Selection for Machine Learning. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/a-gentle-introduction-to-model-selection-for-machine-learning/

[4] Arabian, H., Wagner-Hartl, V. & Moeller, K. (2022). Network Architecture Influence on Facial Emotion Recognition. *Current Directions in Biomedical Engineering*, *8*(2), 524-527. https://doi.org/10.1515/cdbme-2022-1134

[5] Saxena, S. (2023). Introduction to The Architecture of Alexnet. *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2021/03/introduction-to-the-architecture-of-alexnet/

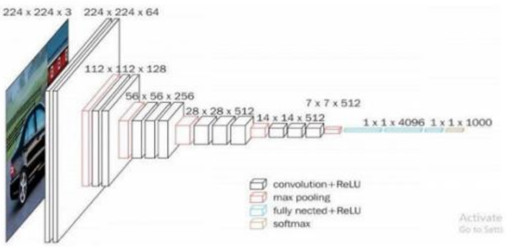
# Appendix

## 1. Appendix 1

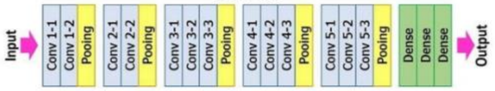


## 2. VGG16 Architecture

VGG16 Architecture

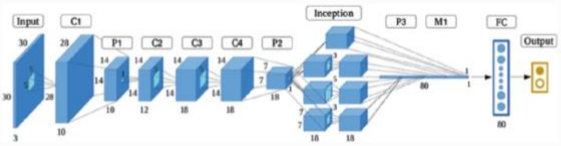


VGG-16



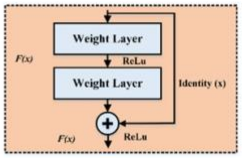
## 3. GoogleNet Architecture

Fig. 1.

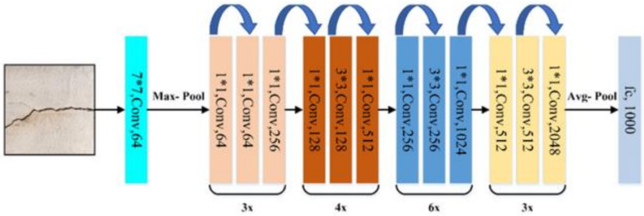


Basic architecture of GoogLeNet

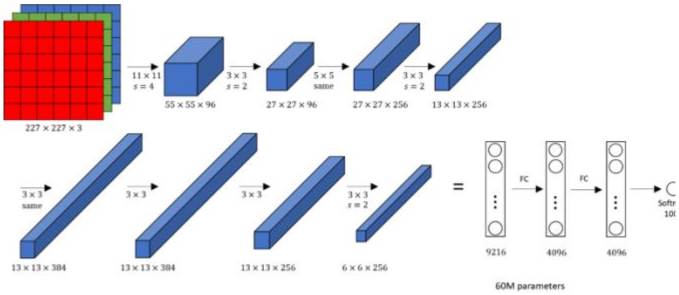
## 4. ResNet Architecture



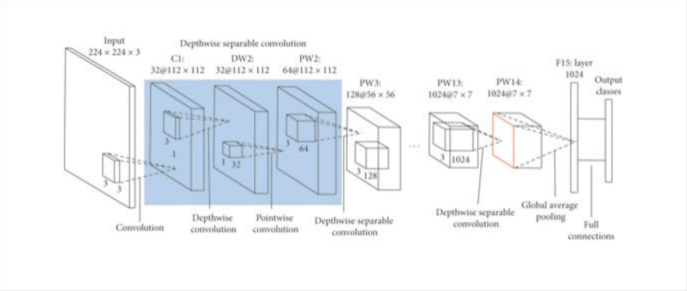
Residual Learning Block



## 5. AlexNet architecture



## 6. MobileNet Architecture


Input
224 × 224 × 3
Depthwise separable convolution
C1:
32@112 × 112
DW2:
32@112 × 112
PW2:
64@112 × 112
PW3:
128@56 × 56
PW13:
1024@7 × 7
PW14:
1024@7 × 7
F15: layer
1024
Output
classes
3
3
3
1
1
32
3
64
3
128
3
1024
Convolution
Depthwise
convolution
Pointwise
convolution
Depthwise separable
convolution
Depthwise separable
convolution
Global average
pooling
Full
connections