Face Emotion Recognition

CNN VS MobileNetV2

Mohamad Ali GHADDAR

Ali ABDALLAH

Juste BANKUMUKUNZI

# **Introduction:**

Emotion recognition from facial expressions plays a significant role in understanding human behavior and improving human-computer interaction. With the advancements in deep learning, Convolutional Neural Networks (CNNs) have been widely used for this task due to their ability to learn complex features from images. Recently, MobileNetV2, a lightweight deep learning architecture optimized for mobile and embedded vision applications, has gained attention for its efficiency and effectiveness in various computer vision tasks. In this report, we compare the performance of CNN and MobileNetV2 models for face emotion recognition.

# **Dataset:**

The project utilizes the FER2013 (<https://www.kaggle.com/datasets/msambare/fer2013>) dataset, which is noted for its diversity and suitability for training deep learning models. This dataset includes a wide range of facial expressions across different demographics, making it ideal for developing robust facial emotion recognition systems.

The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image.

We categorized each face based on the emotion shown in the facial expression into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The training set consists of 28,709 examples and the public test set consists of 3,589 examples.

# **Preprocessing Techniques:**

To enhance the quality of the dataset, various preprocessing steps were undertaken before training. These included resizing the images, performing data augmentation to introduce variability, and splitting the data into training and validation sets to ensure the model's generalizability.

**First Model:**

# **Model Architecture:**

A sequential CNN was chosen for the task, which is particularly effective for image-related tasks. The model includes several layers designed to automatically learn and extract features from the images, which includes:

* **Padding** = The padding parameter of the Keras Conv2D class can take one of two values: 'valid' or 'same'. Setting the value to “valid” parameter means that the input volume is not zero-padded and the spatial dimensions are allowed to reduce via the natural application of convolution.
* **Activation** = relu :The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.
* **Maxpooling** = Maximum pooling, or max pooling, is a pooling operation that calculates the maximum, or largest, value in each patch of each feature map. The results are down-sampled or pooled feature maps that highlight the most present feature in the patch, not the average presence of the feature in the case of average pooling.
* **Batch normalization** = Batch normalization is a technique for training very deep neural networks that standardizes the inputs to a layer for each mini-batch. This has the effect of stabilizing the learning process and dramatically reducing the number of training epochs required to train deep networks.
* **Dropout** = Dropout is a technique used to prevent a model from overfitting. Dropout works by randomly setting the outgoing edges of hidden units (neurons that make up hidden layers) to 0 at each update of the training phase.
* **Adam** = Adaptive Moment Estimation is an algorithm for optimization technique for gradient descent. The method is really efficient when working with large problems involving a lot of data or parameters. It requires less memory and is efficient. Intuitively, it is a combination of the ‘gradient descent with momentum’ algorithm and the ‘RMSP’ algorithm. It uses the squared gradients to scale the learning rate like RMSprop and it takes advantage of momentum by using the moving average of the gradient instead of the gradient itself like SGD with momentum.

# **Training and Evaluation:**

The model was trained on 100 epochs, achieving an accuracy of 96% on the training set and 64% on the validation set. These results were visually represented through accuracy and loss plots, alongside a confusion matrix to evaluate the model's performance for emotions between reality and the model’s predictions.

# **Results:**

The CNN model demonstrated strong predictive capabilities for most emotions with an overall accuracy of 62.5% across all classes (5/8). However, it showed comparatively lower performance in predicting 'angry' and 'fear' emotions, which could be attributed to the lesser amount of data available for these classes. This highlights the need for more balanced datasets in training for improved model performance.

**Second Model:**

1. **Model Architecture:**

A sequential MobileNetV2 was chosen for the task, which is particularly effective for image-related tasks. The model includes several layers designed to automatically learn and extract features from the images, which includes:

* **Input Shape:** The model starts with an input layer designed to handle grayscale images of size 48x48 pixels, with a single channel.
* **RepeatChannels Layer:** This custom layer repeats the single grayscale channel three times to convert it to a three-channel image, compatible with MobileNetV2.
* **MobileNetV2 Base Model:** MobileNetV2, pretrained on ImageNet and with its top layers removed, serves as the feature extractor. The layers of MobileNetV2 are frozen to prevent them from being restrained during the training process.
* **Flatten Layer:** This layer flattens the output from the MobileNetV2 into a one-dimensional tensor, making it suitable for the fully connected layers.
* **Dense Layer with ReLU Activation:** A fully connected layer with 256 units using ReLU activation, which helps in learning non-linear combinations of the features extracted by the MobileNetV2.
* **Batch Normalization:** This technique standardizes the inputs to the dense layer, stabilizing the learning process and improving training efficiency.
* **Dropout:** A dropout rate of 30% is applied to prevent overfitting by randomly setting a fraction of input units to zero during training.
* **Dense Layer with ReLU Activation:** Another fully connected layer with 512 units using ReLU activation, further refining the learned features.
* **Batch Normalization:** Additional batch normalization to maintain the stability of the learning process.
* **Dropout:** Another dropout layer with a 30% rate to continue to guard against overfitting.
* **Output Layer:** A dense layer with 7 units and softmax activation, providing the final classification into one of 7 possible classes.
* **Adam Optimizer:** The model uses the Adam optimizer, which is efficient for large datasets and combines the advantages of both 'gradient descent with momentum' and 'RMSprop' algorithms, adapting the learning rate and improving convergence.

# **Training and Evaluation:**

The model was also trained on 100 epochs, achieving an accuracy of 96% on the training set and 42% on the validation set. These results were visually represented through accuracy and loss plots, alongside a confusion matrix to evaluate the model's performance for emotions between reality and the model’s predictions.

# **Results:**

The MobileNetV2 model demonstrated strong predictive capabilities for most emotions with an overall accuracy of 12.5% across all classes (1/8). It showed comparatively lower performance in predicting both emotions especially in predicting “angry”, “fear” and “sad”. This could be attributed to the lesser amount of data available for these classes. This highlights the need for more balanced datasets in training for improved model performance.

**Conclusion:**

The two models highlighted the challenges of emotion recognition, particularly the need for balanced data and the trade-offs between model complexity and computational efficiency. The results suggest that while MobileNetV2 offers advantages in speed and resource usage, it may require further tuning and potentially more sophisticated data augmentation techniques to improve its performance in real-world applications. Future work should explore the integration of these models with additional modalities, such as contextual and physiological data, to enhance the accuracy and applicability of emotion recognition systems in diverse settings.

***THANK YOU***