**Facial Emotion Detection Using Convolutional Neural  
Networks and Representational Autoencoder Units**

**Group number – 14**

**Rami Amasha**

**Ahmad Bsese**

**Problem definition**

Facial emotion detection is the process of identifying the emotions expressed on the face. Its important in fields such: psychology, marketing and human-computer interactions.

**Project goal:**

The main goal of this project is to develop an approach for facial emotion detection using CNN &Autoencoders.

**Previous approaches:**

Always it was an interesting field " facial emotion detection ", such as :

1. Rule-Based systems: rely on predefined rules or heuristics to identify emotions.
2. SVMs: by finding the hyperplane in high dimensional space that maximally separates different classes.

**CNNs & Autoencoders:**

* CNN : are deep learning models that well-suited for image classification.
* Autoencoders: are neural network that can be used to learn efficient representation of data.
* Autoencoders can be trained on the same dataset as the CNN and it could improve its performance by reducing the dimensionality of the input.

**What is CNN?**

* It is type of neural network that well suited for dealing with data like images.
* They are composed of a series of layers that extract features from the input data.(layers at the beginning extract simple features and deeper layers extract more complex features).

**What is representational autoencoder units ?**

* They are type of artificial neural network that can learn a low-dimensional representation of dataset.
* They consist from two parts :
* Encoder: map the data to a low-dimensional representation.
* Decoder: map back the encoded data to the original dimension.
* The low-dimensional encoding learned by the autoencoder can be used for tasks such as dimensionality reduction, feature extraction, and data visualization.

**Methodology:**

**JAFFE dataset:**

It consists from 213 images of Japanese female displaying 7 basic emotions.

Angry ,happy ,disgust ,surprised ,fear ,neutral ,sad.

**Preprocessing steps:**

As part of preprocessing steps we did data augmentation:

* For each image we have generated 16 batches with size 48x48.
* It's important to consider the trade-off between image size and model performance when choosing the size of input images.

**CNN Architecture:**

1. ***Input layer:***

🡺 they are responsible for learning features from input data.

🡺 They perform a dot product between the filters and small region of the input data and apply an activation function to the result.

1. ***Pooling layer:***

🡺this layer is used to reduce the size of the feature maps.

🡺there are several types of pooling, such as : max pooling and average pooling.

1. ***Fully connected layers:***

🡺they are used to make predictions on the learned features.

🡺 They take the flattened feature maps from the previous layer as an input and apply a dot product with the weights and an activation function to produce the output.

1. ***Output layer:***

🡺its responsible for producing the output of the network.

🡺number of nodes in the output layer is the same as the number of classes we are trying to predict.

**Training procedures:(CNN & Autoencoder)**

1. The autoencoder is first trained reconstructing the input image and then the learned weights are used as the intimal weights for the classifier network, which is trained on the task of classifying the images by emotions.
2. The encoder part of the autoencoder is used as the feature extractor, which has learned useful representations of the input images during the training process of the autoencoder.

* We have trained the autoencoder, and used the encoder part as an input for the CNN with weight that he have learned, and then train the CNN model.

**Results :**

🡺 Results for testing the CNN & Autoencoder model on JAAFE dataset:

* Test Loss: 0.10. 2) Test Accuracy: 63.65
* Results for testing the model on unseen images :
* Test Loss on fer2013 dataset: 0.17
* test accuracy on fer2013 dataset: 48.9

🡺 Results for testing the CNN model on JAFFE dataset:

* Number of Correct Predictions: 799
* Total number of Images: 864
* Accuracy: 92.4

🡺Results for testing the CNN model on fer2013 dataset :

* Number of Correct Predictions: 2942
* Total number of Images: 3,589
* Accuracy 81.97

**Conclusions:**

* The CNN “individual” performed better than the combined model “ autoencoder and CNN” and for this might be many causes, like:

1. **Overfitting**: The combined model may be more prone to overfitting, as it has more parameters to train, and the autoencoder may be learning features that are not relevant for the final task.
2. Feature representation: The features learned by the autoencoder may not be as useful for the final task as those learned by the CNN. The CNN may be better at extracting task-specific features directly from the input, rather than first compressing the input and then extracting features.
3. **Data Quality**: The quality of the data could also influence the results, the CNN model could be benefitted from better quality or amount of data.