Capstone Project: Facial Emotion Recognition

Final Report

Executive Summary

This project proposes a Deep Learning model for detecting human emotions on images of facial expressions. The model is able to perform multi-class classification (happy, sad, surprise and neutral) on images to classify the expression according to the associated emotion. Different models were explored during the project including some pretrained models such as VGG16. The final model boasts very good performance in identifying some classes ('happy', 'surprise') and moderate one in the other emotions ('sad', 'neutral').

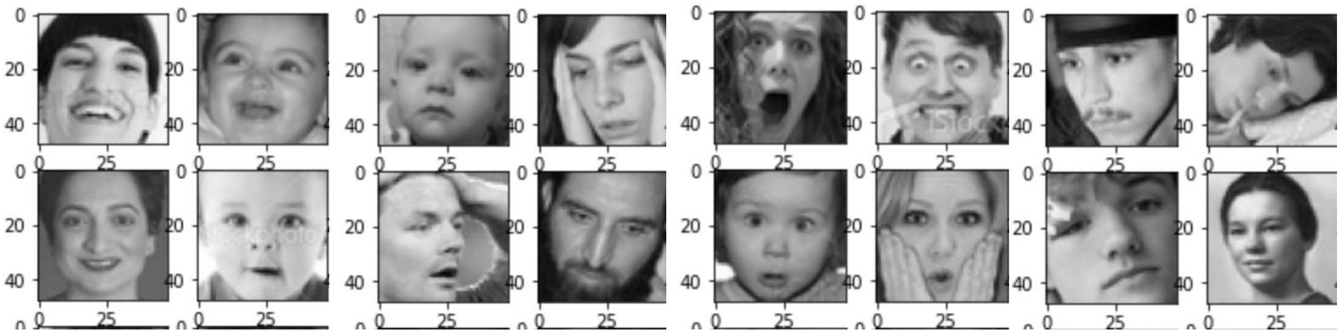
Problem Summary

Deep Learning has found more and more applications in many predictive tasks relating to unstructured forms of data (images, text, audio and video) over the last few years. Many of these tasks aim to match human-level performance in performing intelligent actions on such unstructured data.

Artificial Emotional Intelligence is a specific branch of AI that stands for the study and development of technologies and computers that can read human emotions by means of analyzing body gestures, facial expressions, voice tone, etc. and react appropriately to them. In the field of human-machine interaction, **facial expression recognition is critical**. From recent research, it has been found that as much as **55% of communication of sentiment** takes place **through facial expressions** and other visual cues. Therefore, training a model to identify facial emotions accurately is an important step towards the **development of emotionally intelligent behavior in machines** with AI capabilities. Automatic facial expression recognition systems could have many applications, including but not limited to the detection of mental disorders, or creating a higher quality of virtual assistant for customer-facing businesses.

Solution design

The data used to train and validate the model consisted of 3 folders of labeled images respectively used for: test, train and validation. The images are RGB "white and black" images of size 48x48 pixels.



We have 20K images in total with a 75/25 training/validation split and around 130 images reserved for testing. However, these images are not equally distributed among all classes, as we can see in the following histogram the 'surprise' class is underrepresented compared to the other three:

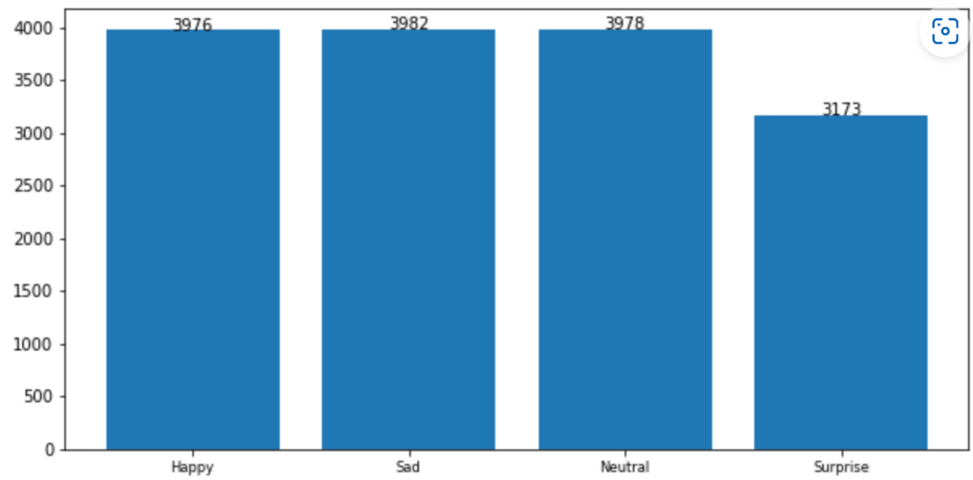
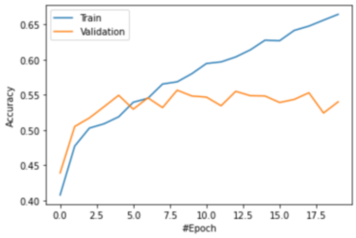
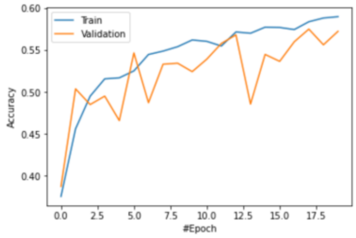
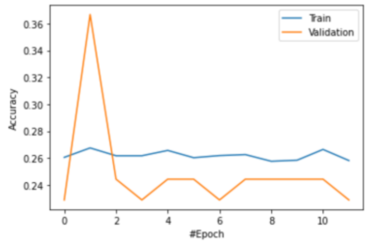


Figure 1: Distribution of image classes

To overcome this imbalance in the dataset, we used data augmentation techniques and Keras image data loaders to train and validate the different models.

A number of different CNN models and methods were explored as part of the solution design, including 3 different pre-trained models: VGG16, ResNetV2, EfficientNet. CNNs models are very effective in reducing the number of parameters without losing the main information. They are commonly used for unstructured forms of data such as images because of the high dimensionality of the data. However, CNNs often need huge amounts of data to train (costly) and the architecture is more complex to design (research and time effort). This is why we used transfer learning techniques to explore some pretrained models on our training data. The performance of the these models is represented in the following table:

VGG	RestNet	EfficentNet
		
14,7M trainable parameters	42,7M trainable parameters	8,7M trainable parameteres
Random model (~50% accuracy) Becomes overfitted around epoch 7	Random model too (similar performance to VGG16) Slightly overfitted	Uncertain model (training accuracy not increasing)

The final proposed solution is a convolutional neural network model composed of five CNN blocks and two fully connected layers used for later classification. The architecture of the final model is represented in the following image:

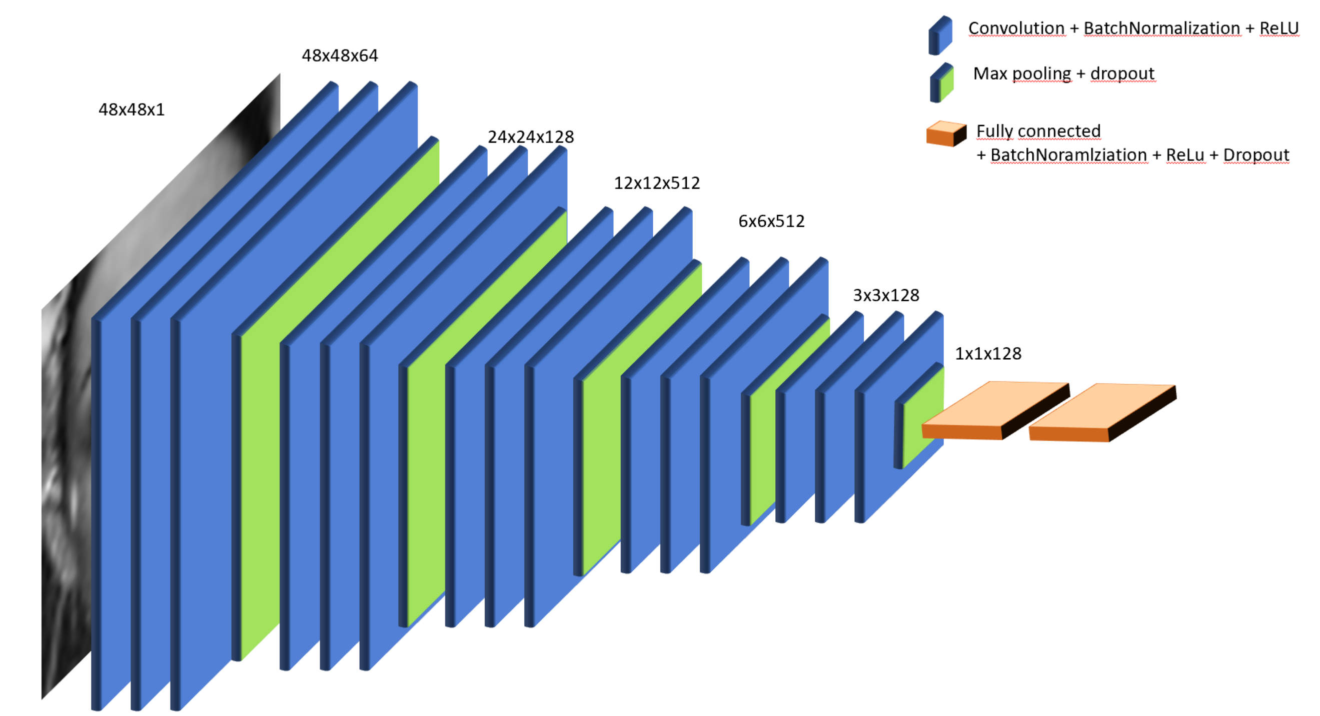


Figure 2: Model's CNN architecture

Figure 3 shows the model's training and validation accuracy increases with the different epochs during the training phase. The final outcome is a slightly overfitted model (validation accuracy: 72%, training accuracy: 76%) with very good performance in comparison with previous models.

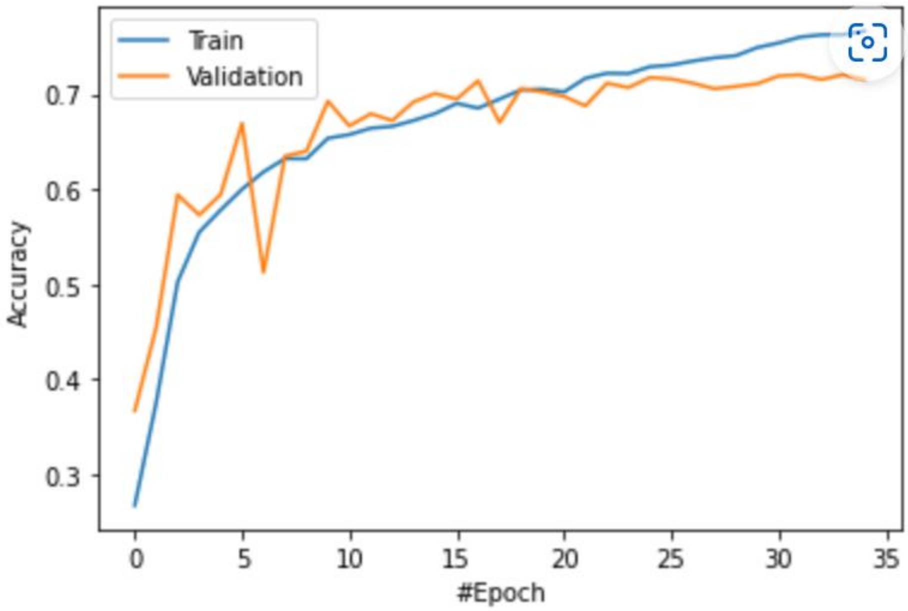


Figure 3: Model training - Accuracy keeps increasing with number of epochs

Analysis and Key Insights

We used the 128 remaining images to test model performance on a new dataset. The classification report in **Figure 2** shows the model's accuracy is overall good with a 78% accuracy (top-5 accuracy 91%). The model has higher performance for happy and surprise emotions recognition than for neutral or sad. **Figure 3** illustrates for the confusion matrix representing for each class or emotion, the number of times it was correctly identified and misinterpreted as other emotions. The most common mistake is classifying a 'neutral' face as 'sad'.

	precision	recall	f1-score	support
0	0.83	0.83	0.83	12
1	0.67	0.80	0.73	5
2	0.67	0.57	0.62	7
3	0.88	0.88	0.88	8
accuracy			0.78	32
macro avg	0.76	0.77	0.76	32
weighted avg	0.78	0.78	0.78	32

Figure 4: Classification report

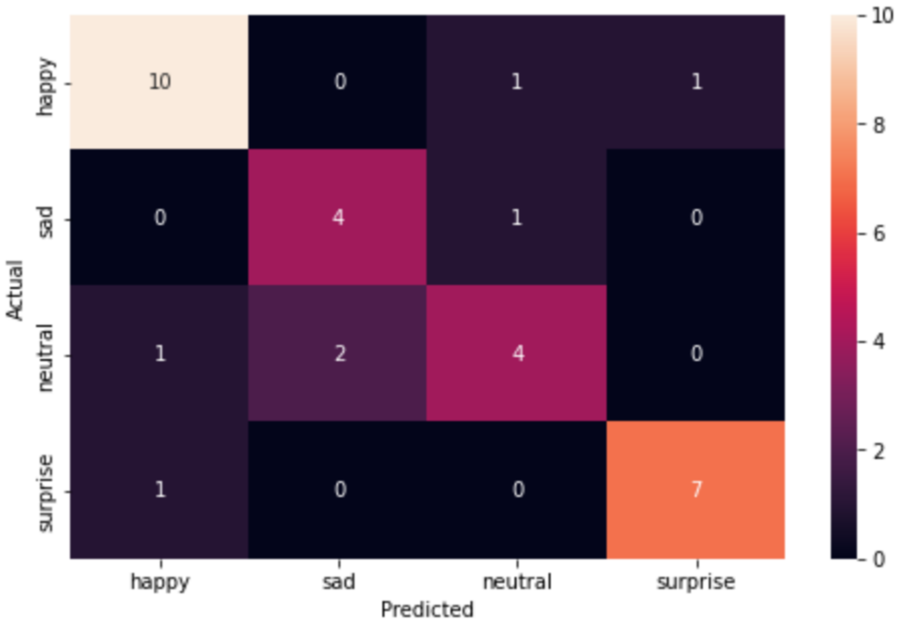


Figure 5: Confusion matrix

Limitations and Recommendations for Further Analysis

For the purpose of facial emotion recognition, stakeholders need to consider whether some of the emotions are more important to be recognized than others. For example, if the end application involves interacting with depression recovering patients, in this case it is more important for the model to effectively recognize a sad emotion over other emotions. If this is the case, we would need to change the model approach to develop a model that prioritizes sad classification recall over other evaluation metrics.

The model's performance is generally good, but there are many things to explore in the future that could improve performance:

- Data augmentation: CNN models require huge amounts of data to learn the patterns on the image. This model was trained with 20K images which is a relatively small size, and the 'surprise' emotion is underrepresented in the data. Increasing the size of the dataset is highly probable to improve model performance.

  The process of data augmentation for image labeled datasets can be tedious and time consuming (labeling images). Another option that could be used instead to try to improve model performance is to explore new data augmentation techniques such as rotating or cropping the images.

- Transfer learning of new algorithms: In this project, we tried using the complete architecture of the pretrained models as feature extraction baseline. In next steps, it would be interesting to check the impact of using only some of the layers of the pretained models together instead.

Bibliography

- Tang, Tanya 2020. Dynamic Emotion Detector using Transfer Learning.Dynamic Emotion Detector using Transfer Learning | by Tanya Tang | The Startup | Medium
- Chobanyan, Martin. 2019. Training an emotion detector with transfer learning. Training an emotion detector with transfer learning | by Martin Chobanyan | Towards Data Science
- Azeem Rao, Muhamed. 2021. Realtime Face Emotion Recognition using transfer learning in TensorFlow. Realtime Face Emotion Recognition using transfer learning in TensorFlow | by MUHAMMAD AZEEM RAO | Analytics Vidhya | Medium
- Singh Minhas, Manpreet. 2019. Image Data Generators in Keras. Image Data Generators in Keras. How to effectively and efficiently use… | by Manpreet Singh Minhas | Towards Data Science
- Browniee, Jason. 2019. How to Configure Image Data Augmentation in Keras. How to Configure Image Data Augmentation in Keras (machinelearningmastery.com)