Final Submission: Facial Emotion Detection

Yonghoon Yoon

Executive Summary

This project explores different models for detecting facial emotion from images, with the Complex Neural Network Architecture being the best-performing and proposed solution for facial emotion detection. Such models can improve the experience and quality of E-Learning, as they can track the level of engagement and understanding from students. The proposed model contains five convolutional layers and 2 fully connected layers, with other layers that help reduce internal covariate shift and overfitting. It uses ReLU and LeakyReLU as activation functions to help with the gradient descent.

It is to be noted that our dataset contains some non-human faces or objects, and the test dataset is very small compared to the train and validation sets. There is also an imbalance between the emotion labels, which means that we will focus on the f-1 score as the evaluation metric.

The model performs with a high f1 score on the four emotions (happy, sad, neutral, and surprise) but would be better if we had larger and more diverse dataset, and labeled into 7 emotions like one of the models. (happiness, sadness, disgust, neutral, fear, anger, surprise) but the model will help to detect emotions to personalize e-learning.

While efficient facial emotion detection models can help improve E-Learning, there are concerns relating to privacy and transparency. Not only do these concerns affect the legality of the application of this technology, but they also affect the collection of diverse and quality data. Without the consent of students during their lessons, it will be hard to obtain enough quality data that will help train our model best. Hardware and software requirements must also be met, such as 16 of RAM and a GPU unit.

It is recommended that the E-Learning implementation businesses improve and use facial emotion detection to further the impact of E-Learning in terms of accessibility and quality. It is crucial, though, that they use it carefully, especially regarding the privacy and transparency of their clients.

Problem Summary

With the emergence of online tools like YouTube and Chegg, online education, or E-Learning consistently evolved at all levels of education, particularly in higher education. Then came the COVID-19 outbreak, which significantly increased the usage, consideration, and investment in E-Learning. Such a phenomenon demanded more accessible and quality online education, especially with its biggest flaw being the lack of personability factor through the screens. This is where deep learning can help E-Learning become better in accessibility and quality, through emotion detection.

Emotion detection can have different methods, such as brain signal processing (e.g., EEG), voice/speech processing, facial movement processing, and text processing. By detecting the emotions of students, online instruction can improve in terms of focus retention and content understanding. This is true for both human instructors and AI instructors; here are some examples: a prompt to take a stretch or a quick engaging game can be suggested if the student seems bored or tired. Another type of prompt can be used to review a specific topic again when the student seems confused.

To that end, this project uses a dataset that contains images of four emotions (happy, sad, neutral, and surprised) to create a facial emotion detection model. The model will use a complex convolutional neural network with hyperparameter tuning to arrive at a model that can accurately distinguish the aforementioned emotions. A further application of this model will contribute to detecting the emotions of online learning students and improve the quality of E-Learning as a whole.

Solution Design, Analysis, and Key Insights

Several different models were tested to build the best-performing model in facial emotion detection. They included simpler CNN to complex neural network architecture, along with different transfer learning models using VGG, ResNet, and EfficientNet. The final proposed solution is the complex neural network architecture, with five layers of convolutional layers with two fully connected blocks and hyperparameter tuning. Its validation accuracy of 0.71 and its test accuracy of 0.84 were the highest among the models. Below are the visuals that show the performance of the proposed model.

Here is the summary of how different models performed:

Model Description	Validation Accuracy	Test Accuracy	Weighted AVG f-1 Score
Simple CNN #1	0.68	0.72	0.21
Simple CNN #2	0.68	0.66	0.28
VGG (Transfer)	0.51	0.48	0.32
ResNet (Transfer)	0.37	0.25	0.10
EfficientNet (Transfer)	0.23	0.25	0.10
Complex CNN #1	0.62	0.56	0.52
Complex CNN #2	0.65	0.72	0.71
Complex CNN #3	0.64	0.72	0.71
Complex CNN #4	0.71	0.84	0.78
Complex CNN #5	0.72	0.84	0.78

Figure 1: Performance across different models

Figure 2 shows the training process of the proposed model. There is not too much evidence of overfitting thanks to the dropout and batch normalization layers. There is some noise in the trend though.

The validation accuracy ranged from 0.28 to 0.72 throughout the process

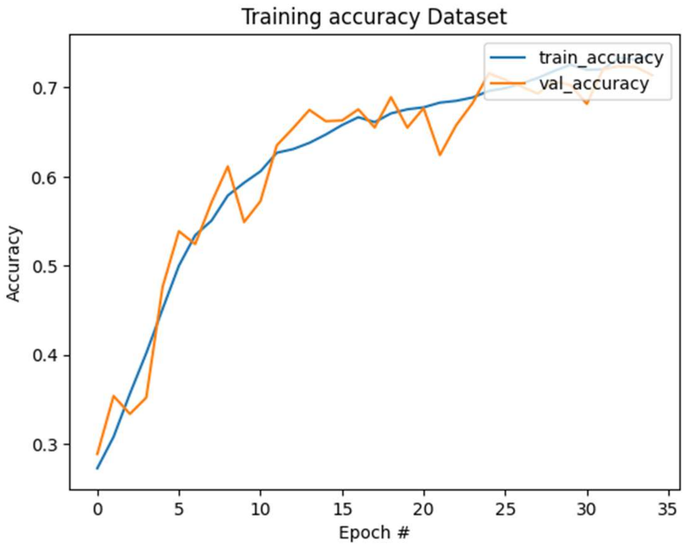


Figure 2: Training and validation accuracy through the epochs

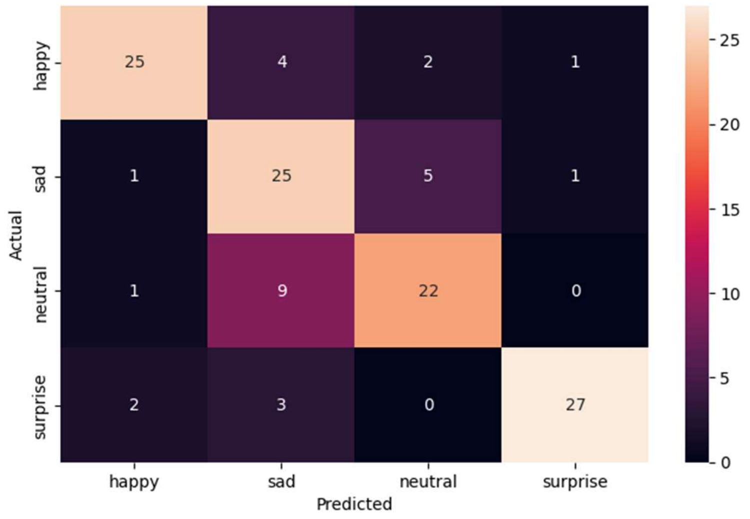
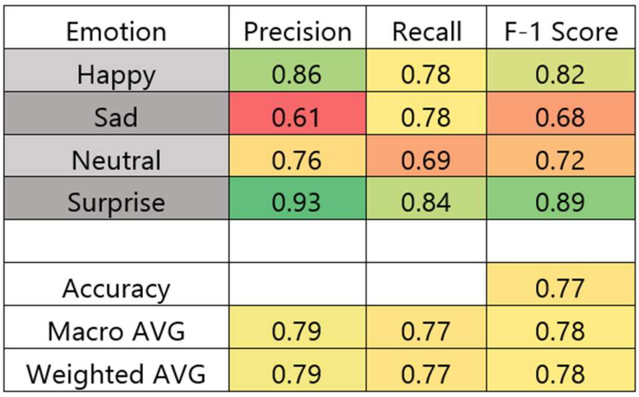


Figure 3: Predicted labels compared to actual labels

Figure 3 shows that the model correctly predicted 22-27 images out of 32, of which the best performance was for 'surprise' images. This is likely because there are most distinguishable features in surprise images in terms of big eyes and moths, and many times, hands on the faces. Neutral images had the lowest correct predictions, while the model had the most incorrect 'sad' predictions.

Figure 4 illustrates a precision, recall, and F-1 score for each class. Due to the imbalance in the data, we want to focus on the F-1 score, which is the weighted harmonic mean of precision and recall. As we expected from the above heatmap, the class of surprise had the highest F-1 score and sad and neutral images had the lowest F-1 scores. The weighted AVG F-1 score is 0.78, which is the highest among the models that were explored.



Emotion	Precision	Recall	F-1 Score
Happy	0.86	0.78	0.82
Sad	0.61	0.78	0.68
Neutral	0.76	0.69	0.72
Surprise	0.93	0.84	0.89
Accuracy			0.77
Macro AVG	0.79	0.77	0.78
Weighted AVG	0.79	0.77	0.78

Figure 4: precision, recall, and f-1 score for the classes

The following approaches were applied to optimize the model:

- The learning rate for the Adam optimizer was reduced to 0.001 to avoid exploding gradient descent.
- A batch size of 32 was used to reduce the memory while converging faster than stochastic gradient descent where the batch size is 1.
- The cross entropy loss function was used to determine how well the model fits the dataset.
- The number of epochs was increased especially due to the slow learning rate.
- Dropout layers were added to prevent overfitting, and the rate was reduced to 0.1.
- Batch Normalization helped stabilize the model.
- The grayscale color_mode was used because it reduced training time due to the lower dimensions of the model input.

With further improvement, the model should be able to perceive students'

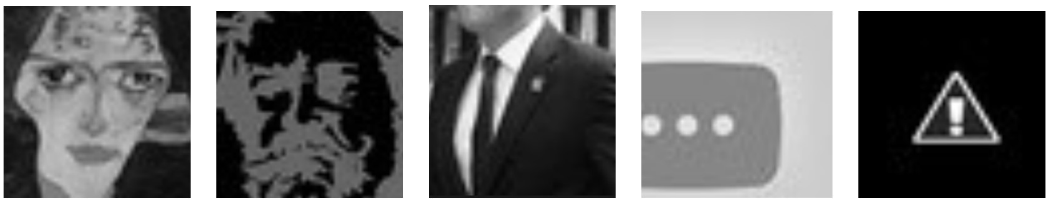
emotions better, and even other emotions like fear, disgust, and anger. With the combination of these detections, new models can be made that can gauge when students are bored, confused, or need social-emotional attention. Furthermore, this type of technology can be applied to AI instructors as well, which can cut costs and improve customer satisfaction as well.

Limitations and Recommendations for Further Analysis

Even though the complex CNN architecture was optimized with the above-mentioned tuning approaches, the weighted average f-1 score was still 0.78.

Here are some of the potential reasons for the low performance and recommended solutions for them:

- The images were small, grayscale, and many of them blurry. Having a **larger**, **more diverse**, and **quality** dataset would help with the f-1 score.
- There were some non-facial images along with some drawings that made it hard to tell the emotion (shown below). Such data would confuse the training and validation process, so **refining the content** of the data would be helpful.



- Adding **more emotion classes** and tuning our dataset to include more **clear** examples of each class would improve the model. For example, this model used seven emotion classes (happiness, sadness, neutral, fear, disgust, surprise, anger).

Eventually, models should include which combination of emotions are exhibited when students are bored, confused, or need social-emotional attention. Then, models that detect emotions via different methods (i.e. voice, gesture, and text) can be integrated to improve the model overall.

Recommendations for Policy and Implementation

The above analysis shows that models can be improved to detect facial emotions, especially with a larger dataset with better quality and diverse images. It is recommended that the stakeholders invest in further development of this model to evolve E-Learning to the next level.

Here are some costs and concerns to consider:

- There is a wide concern around **privacy and transparency** when it comes to training models to detect facial emotion. This is one of the most critical issues for development, so stakeholders should navigate the **legalities and ethics around collecting data and developing it**, so it can only better the society.
- For this project, about 8 GB of RAM and a T4 GPU unit was necessary. Even with these resources, the training process was not time efficient, so it is recommended that **hardware with higher specifications** be used.
- Stakeholders and data scientists involved should also **look into other models** such as the model developed by Tanoy Debnath, Md. Mahfuz Reza, Anichur Rahman, Amin Beheshti, Shahab S. Band and Hamid Alinejad-Rokny for inspiration and transfer learning.
- As discussed in the previous section, further research should include which emotions are important in monitoring student mood, such as boredom, confusion, and stress.

Bibliography

Bouchrika, I. (2024). 66 eLearning Statistics: 2024 Data, Analysis & Predictions. *Research.com*. Retrieved February 21, 2024, from https://research.com/education/elearning-statistics

Debnath, T., Reza, M. M., Rahman, A., Beheshti, A., Band, S. S., & Alinejad-Rokny, H. (2022). Four-layer ConvNet to facial emotion recognition with minimal epochs and the significance of data diversity. *Scientific Reports*, *12*(1). https://doi.org/10.1038/s41598-022-11173-0

How Emotional Artificial intelligence can Improve Education | Edlitera. (n.d.). Edlitera. https://www.edlitera.com/blog/posts/emotional-artificial-intelligence-education

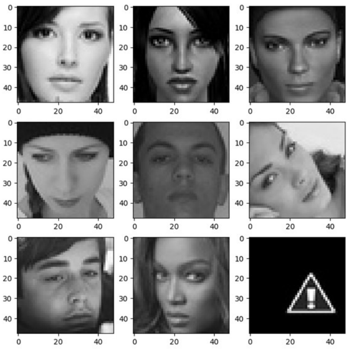
Intetics Inc. (2023, April 27). *Education Software Solutions: elearning software development, educational software development - Intetics*. Intetics. https://intetics.com/industries/education-and-elearning/

Machová, K., Szabóová, M., Paralič, J., & Mičko, J. (2023). Detection of emotion by text analysis using machine learning. *Frontiers in Psychology*, *14*. https://doi.org/10.3389/fpsyg.2023.1190326

Tesler, I. (2022, October 8). *How AI Facial Recognition and Emotion Detection Helps Businesses. Check Yourself with Fun Demo*. Intetics. https://intetics.com/blog/how-ai-facial-recognition-and-emotion-detection-helps-businesses-check-yourself-with-fun-demo/

Appendix

Appendix 1: A sample of “neutral” images, including a non-facial image mentioned above.



Appendix 2: Complex Neural Network Architecture Layers in Python

```
no_of_classes = 4

model3 = Sequential()
```

```
# 1st CNN Block
model3.add(Conv2D(filters = 64, kernel_size = 2, padding = "same", input_shape
= (48, 48, 1), activation = 'relu'))
model3.add(BatchNormalization())
model3.add(LeakyReLU(alpha = 0.1))
model3.add(MaxPooling2D(pool_size = 2))
model3.add(Dropout(0.1))

# 2nd Conv2D layer with 128 filters and a kernel size of 2. Use the 'same' padding and 'relu'
activation.
model3.add(Conv2D(filters = 128, kernel_size = 2, padding = "same", activation = 'relu'))


# Third Conv2D layer with 512 filters and a kernel size of 2. Use the 'same' padding and 'relu'
activation.
model3.add(Conv2D(filters = 512, kernel_size = 2, padding = "same", activation = 'relu'))
model3.add(BatchNormalization())
model3.add(LeakyReLU(alpha = 0.1))
model3.add(MaxPooling2D(pool_size = 2))
model3.add(Dropout(0.1))

# Fourth block, with the Conv2D layer having 512 filters.
model3.add(Conv2D(filters = 512, kernel_size = 2, padding = "same", activation = 'relu'))

# Fifth block, having 128 filters.
model3.add(Conv2D(filters = 128, kernel_size = 2, padding = "same", activation = 'relu'))
model3.add(BatchNormalization())
model3.add(LeakyReLU(alpha = 0.1))
model3.add(MaxPooling2D(pool_size = 2))
model3.add(Dropout(0.1))
# Flatten layer, followed by Dense layers.
model3.add(Flatten())

# First Dense layer with 256 neurons followed by a BatchNormalization layer, a 'relu' Activation,
and a Dropout layer. This forms the first Fully Connected block
model3.add(Dense(256, activation = 'relu'))
model3.add(BatchNormalization())
model3.add(ReLU())
model3.add(Dropout(0.5))

# Second Dense layer with 512 neurons, again followed by a BatchNormalization layer, relu
activation, and a Dropout layer.
model3.add(Dense(512, activation = 'relu'))
model3.add(BatchNormalization())
model3.add(ReLU())
model3.add(Dropout(0.5))

# Final Dense layer with 4 neurons.
model3.add(Dense(no_of_classes, activation = 'softmax'))
```