Face It: A Computer Can Read Your Emotions

Using Convolutional Neural Networks (CNNs) To Interpret Facial Expressions

Abstract/Problem Statement

There are two main methods of communication – verbal and nonverbal. According to a study conducted by a UCLA psychology professor, Albert Mehrabian, these nonverbal methods of communicating may be even more important at conveying meaning than the actual words you say. He is credited with the 7-38-55% rule, which establishes that the meaning of a message is only 7% based on the words used and just 38% based on the tone of voice, while more than half (55%) is based on body language and other visual cues, like facial expressions.

Given that visual cues impart such a significant amount of meaning to communication, those who are visually impaired could potentially misinterpret what is being spoken. Consequently, the goal of this project was to create a machine learning model using a CNN that could interpret facial expressions from a video stream. In doing so, the computer could provide an audio output and visual label of a person's emotional state. This would assist those who are visually-impaired by informing them of how someone is feeling while they interact with each other. The model could also be helpful for those on the autism spectrum who may have difficulty interpreting facial expressions and need some assistance. As a result, I believe this model would be significantly useful in helping improve quality of life for a wide demographic of people by promoting positive social interaction.

In applying this model to the real world, there are a few potential obstacles to its accuracy. For example, there are many occasions where people's facial expressions are incongruous with their feelings. The accuracy of the model may also be affected by the lighting in the video. Lastly, now that wearing masks is a new normal during the COVID-19 pandemic, it will be difficult for the model to use the form of a person's mouth in making its decisions.

Data

The model was trained and tested on 2,473 images (from https://doi.org/10.7910/DVN/358QMQ) of people displaying one of the seven basic emotions: happy, sad, angry, fearful, disgusted, surprised, and neutral. The images were taken from video clips of people speaking sign language from the RWTH-PHOENIX-Weather 2014 dataset, and each were labeled with its corresponding emotion. The original weather dataset originated from RWTH Aachen University in German, and the labeled dataset originated from the University of Central Florida. 2,333 of the images were used for training and 140 were used for testing. As these are categorical variables, the data were discrete.

Method

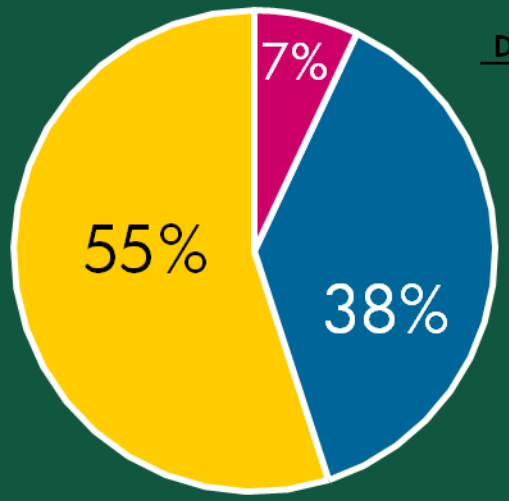
```
model = Sequential([
    Conv2D(64, (3,3), activation='relu', padding='same', input_shape=(48,48,1)),
    MaxPooling2D((2,2)),
    Conv2D(128, (3,3), activation='relu', padding='same'),
    MaxPooling2D((2,2)),
    Conv2D(256, (3,3), activation='relu', padding='same'),
    MaxPooling2D((2,2)),
    Flatten(),
    Dense(256, activation='relu'),
    Dense(64, activation='relu'),
    Dense(7, activation='softmax')
])
model.compile(loss='categorical_crossentropy', optimizer=RMSprop(lr=0.001), metrics=['accuracy'])
history = model.fit(training_gen, steps_per_epoch=(total_train//64), epochs=30,
                    validation_data=testing_gen, validation_steps=(total_test//64))
```

Method Performance

Using the CNN architecture shown above, the model achieved approximately 68% accuracy on the testing set and 93% accuracy on the training set. The testing loss was approximately 1.7 and the training loss was approximately 0.19. This indicates the model is slightly overfit.

An interesting aspect of the model's performance was the occasional delay in output when using live video footage from the webcam. Most of the time, the model was able to determine a person's emotional state relatively quickly. However, at times, the model wouldn't show a new prediction until the person stepped away from the camera and then returned back to the video frame. The model also seemed to predict "Happy," "Angry," and "Neutral" more often than the other emotions.

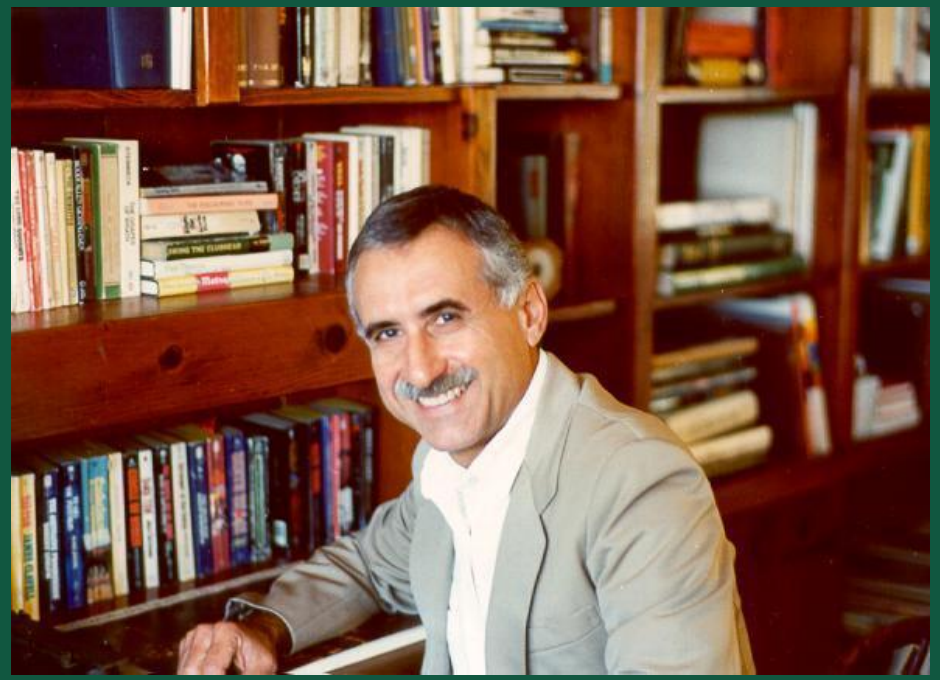
Problem Statement



Dr. Albert Mehrabian's 7-38-55% Rule

Elements of Personal Communication

- 7% spoken words
- 38% voice, tone
- 55% body language



To learn more: https://www.nytimes.com/2006/09/24/books/chapters/0924-1st-peas.html

Training and Testing Datasets

- 2473 total pictures (after data cleanup)
  - 2333 pictures in training, 140 in testing; split across the seven emotion labels
- 486 Angry, 167 Disgusted, 284 Fearful, 159 Happy, 175 Neutral, 305 Sad, 757 Surprise in the training set
  - Potential for errors due to class imbalance
- 20 Angry, 20 Disgusted, 20 Fearful, 20 Happy, 20 Neutral, 20 Sad, 20 Surprise in the testing set
- Some pictures seemed to overlap multiple emotions
- Source: Alaghband, Marie; Yousefi, Niloofar; Garibay, Ivan, 2020, "FePh: An Annotated Facial Expression Dataset for the RWTH-PHOENIX-Weather 2014 Dataset", https://doi.org/10.7910/DVN/358QMQ, Harvard Dataverse, V2, UNF:6:u/2CPrH/I56OfTMqqijTvA== [fileUNF]

Examples of Data

Anger

Disgust

Fear

Surprised

Happy

Neutral

Sad

CNN Model

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 48, 48, 64)	640
max_pooling2d_6 (MaxPooling2	(None, 24, 24, 64)	0
conv2d_7 (Conv2D)	(None, 24, 24, 128)	73856
max_pooling2d_7 (MaxPooling2	(None, 12, 12, 128)	0
conv2d_8 (Conv2D)	(None, 12, 12, 256)	295168
max_pooling2d_8 (MaxPooling2	(None, 6, 6, 256)	0
flatten_2 (Flatten)	(None, 9216)	0
dense_7 (Dense)	(None, 256)	2359552
dense_8 (Dense)	(None, 64)	16448
dense_9 (Dense)	(None, 7)	455

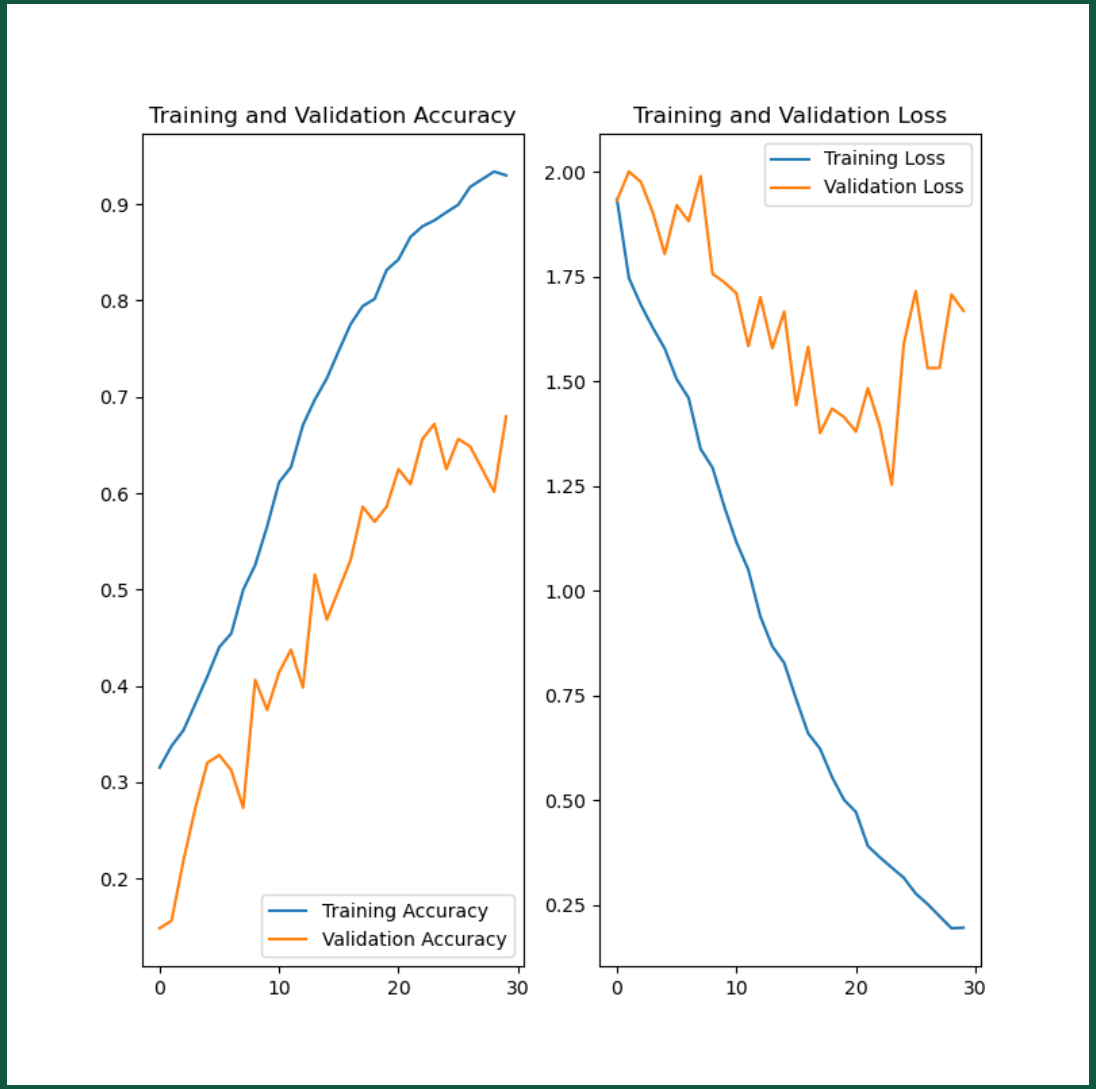
```
model = Sequential([
    Conv2D(64, (3,3), activation='relu', padding='same', input_shape=(48,48,1)),
    MaxPooling2D((2,2)),
    Conv2D(128, (3,3), activation='relu', padding='same'),
    MaxPooling2D((2,2)),
    Conv2D(256, (3,3), activation='relu', padding='same'),
    MaxPooling2D((2,2)),
    Flatten(),
    Dense(256, activation='relu'),
    Dense(64, activation='relu'),
    Dense(7, activation='softmax')
])
model.compile(loss='categorical_crossentropy', optimizer=RMSprop(lr=0.001), metrics=['accuracy'])
history = model.fit(training_gen, steps_per_epoch=(total_train//64), epochs=30,
                    validation_data=testing_gen, validation_steps=(total_test//64))
```

Model Output/Performance

Model Output/Performance

After closing the video stream, the model outputs an audio report of the latest emotion the person displayed. For example:

Model Output/Performance



Future Applications and Further Study

- Future Applications:
  - Online Learning Platforms
  - Marketing
  - Customer Service
    - More powerful when combined with verbal analysis

- Further Study/Potential Improvements:
  - Improved audio integration
  - Facial recognition
  - Include verbal/speech analysis
  - Image augmentation
  - More robust dataset
    - Corrected class imbalance
    - Increased total number of images
    - Distinct display of emotions

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7037130/

Thank You!