Facial Expression Recognition in Remote Learning Conditions

Team6: Malcolm Smith Fraser, Zoe Zhu, Han Gong, Wilson Huang, Tommy Tseng

Abstract

This article presents an application for classifying real-time facial expressions of video conference audiences to better evaluate the overall state of mind in a meeting and improve communication efficiency. To achieve this goal, we employed the real-time computer vision library OpenCV, and convolutional neural network models to extract human faces from live video and classify facial emotions. Accuracy and prediction speed were the two major metrics used to evaluate model performance. It was found that a ResNet50 CNN architecture model scored the highest with an overall accuracy of 72%. Past research [13] found that humans, on average, can accurately classify a static emotion with 65% accuracy on fer2013 dataset. Our model outperformed human baseline classification in a multi-participant Zoom meeting with about 6.44 times faster prediction speed. Though there are some concerns with the model's ability to generalize past the faces in the training dataset, we believe that with more training data this application can help online instructors better identify the state of mind of a class in remote learning contexts.

Introduction

A key part of effective communication is the ability to get quality feedback, both verbal and non-verbal, that can be used to improve engagements future interactions [16]. However, capturing feedback and detecting the real time sentiment of audiences precisely, such as surprise or sadness, can be a challenging task when faces with the grid of faces we frequently see in video conferences.

The goal of this project is to present a final application that can help presenters to better evaluate the overall performance of the meeting attendees and improve overall communication efficiency. Specifically, this application uses a convolutional neural network to detect and classify video conference attendee facial expressions in real-time.

Background

According to Bouhlal et al.[16], the three major phases of a facial recognition system are: the detection phase, the extraction phase, and the classification phase. These three major phases served as our building blocks of our system. In the research conducted by Divjak and Bischof [18], they have used OpenCV to both detect and extract human faces from their own system. Due to it's open-source nature and the ease of use, OpenCV was used to detect, and extract faces in our system.

For the classification phase, we explored three CNN architectures: VGG 16, CNN 10+, and ResNet50. Before ResNet 50 was known to the public, VGG16 had exhibited outstanding predictive performance in the ImageNet competition for a period of time. Plus, pre-trained layers by the University of Oxford are available from the *vggface* package for public use, so the VGG16 was therefore chosen as our baseline model [8]. As a comparison to the pre-trained model, a CNN model constructed by one of the finalists of the Kaggle competition *Challenges in Representation Learning: Facial Expression Recognition Challenge* winners was also included. We named this CNN model "CNN 10+" as the model has 13 layers in its architecture. Finally, as an advanced method of the VGG16, ResNet50 was chosen as our third model because it has been proven to have great prediction performance in past ImageNet challenges [5].

Data

The major dataset we used is the Facial Expression Recognition 2013 (FER-2013) stored as csv file. The dataset consists of 35,887 grayscale, 48 x 48 sized face images with 7 different emotions including happy, sad, angry, disgust, fear, surprise and neutral. The image data is split by space character as a list, resized and augmented to meet the input requirement for each model we fit. The models were trained on 28709 training samples and 3589 validation images and tested on 3589 testing cases.

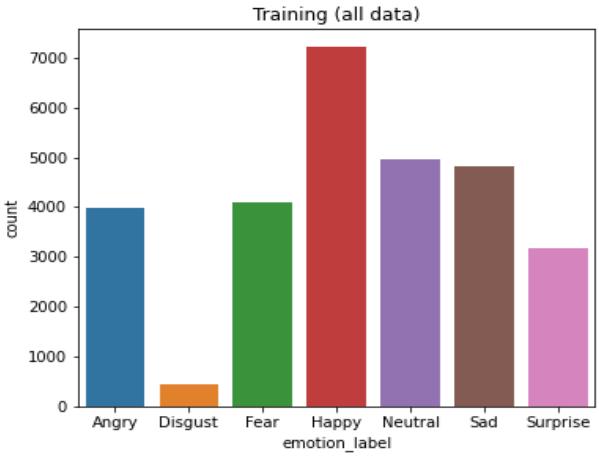

Figure 1. Label distribution for all the fer2013 training data.

Not all the 35887 faces in the fer2013 dataset are able to be detected by OpenCV. This is due to poor lighting, or the face being at an angle that the OpenCV haar cascade face detection algorithm cannot properly process. The underlying class distribution in the dataset, however, is relatively unchanged for the 26482 faces that area able to be detected and thus this issue did not pose a concern in terms of information loss aside from a loss in the total number of overall samples.

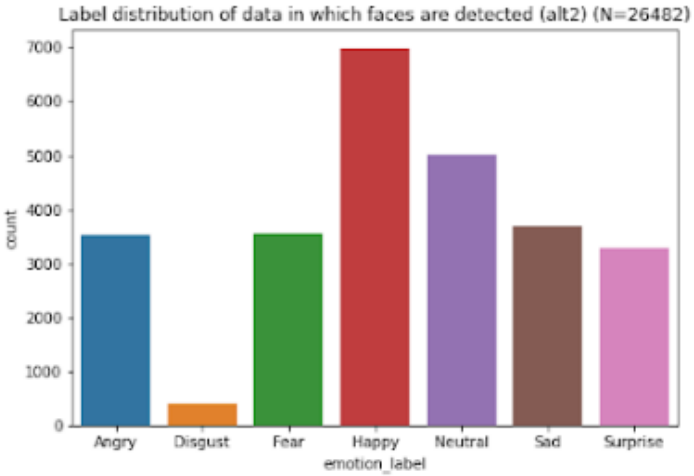

Figure 2. Label distribution of the fer2013 data that can be detected with OpenCV

Upon inspection, many of the faces contained in the dataset are very similar across classes, which is a reminder of the relatively arbitrary nature of facial emotion classification. (*Figure 3*)

Figure 3. Two training images from the fer2013 dataset. The one on the left is labelled neutral, while the one on the right is labelled sad.

Methods

- *OpenCV*

Because the application we planned to build would classify faces extracted using OpenCV, we used OpenCV to extract the faces in the fer2013 dataset as part of the training process. This was only done for the training of the CNN 10+ but not for the transfer learning models due to time and computational capacity constraints. OpenCV was also used to reshape the images and perform any color conversions.

OpenCV is a real-time optimized Computer Vision library. It was selected due to its popularity and ease of use. However, due to some limitations in how OpenCV identifies faces it would be worthwhile for future work to explore other options for face detection.

- *Data Augmentation & Scaling*

Data augmentation techniques are used to increase data diversity, improve model generalizability. Considering possible image transformations under web-camera, we applied horizontal mirroring, ±10-degree rotations, 10% image zooms and 10% horizontal/vertical shifting through *Keras imagedatagenerator* class. In addition to augmentation, we also implemented scaling techniques to normalize pixel values to the range 0-1 for faster learning.

- *Model training*

Models were trained using Google Colab Pro with GPU runtimes. This provided us with an affordable way to significantly speed up the training process. We did face some RAM issues, both during data preprocessing and ResNet50 model training. The only foreseeable solution to avoid such issues would have been to migrate the model training process to a cloud environment (AWS, GCP, Azure, etc) and reserve a more powerful server instance. However, this would have been significantly more expensive and was thus not a feasible option for the team.

- *Transfer Learning*

Transfer learning is a commonly used technique in image detection that allows us to copy the first *n*layers from a base network and trained on the remaining layers of the target network to boost model performance [6]. In this project, we adopted VGG16 and ResNet model from *vggface* package [7] as our base network.

Given the relatively small sample size and significant similarity between our task and the facial recognition task the *vggface* is trained on, we found transfer learning significantly improved our model. *Figure 4* below illustrates the basic concept of transfer learning.

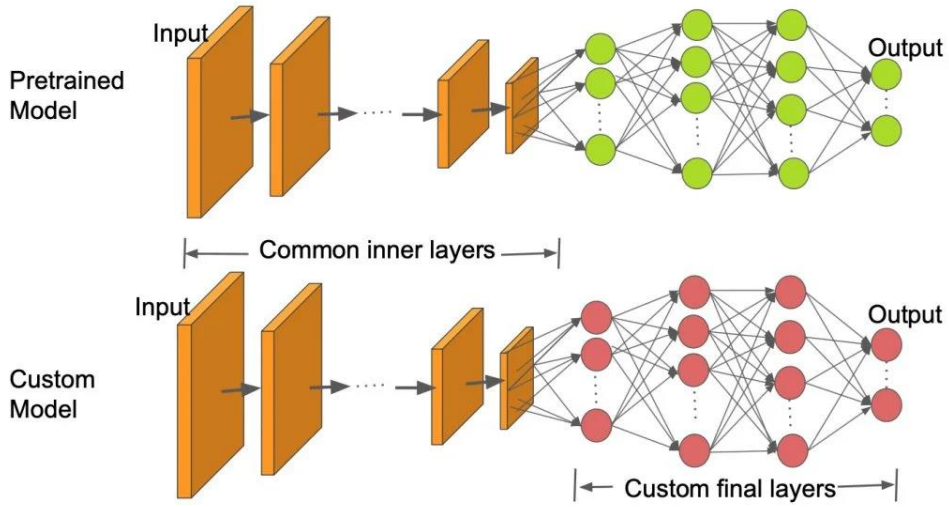

Figure 4. Conceptual representation of transfer learning

- *VGG16 CNN Architecture*

VGG16 was introduced by Karen Simonyan and Andrew Zisserman in 2014 [4]. It contains 16-19 layers and uses multiple 3x3 convolution filters to perform feature selection. It ranked first and second in ImageNet Competition for a period of time before Resnet50 came into the competition. The model requires a 2-dimensional input as in (height, width). Given that the raw data is in (48, 48), *target_size* in the *image* package is specified to modify raw data to appropriate size, (96, 96) in this case. For simplicity and accuracy, we specify the weight to be "imagenet", which indicates a pretrained weight. We finish the rest of the training process by using data augmentation with *file_from_directory* generator. OpenCV is also applied to the training and test dataset, which cropped 16k+ faces out of 28k pictures and 2511 faces out of 3589 pictures. Given that there is no discernible improvement in model accuracy and potential decrease in sample size, openCV is not required in the training process.

- *CNN 10+ architecture*

This CNN was one of the highest scoring models in the *Challenges in Representation Learning: Facial Expression Recognition Challenge* on Kaggle [10]. The 13 layer architecture was implemented using the Keras interface for the Tensorflow library. An Adam stochastic gradient descent optimizer was used with a learning rate of .001, a batch size of 40, and a validation dataset that is 10% of the training dataset. One advantage of this model is that it does not require the input data to be reshaped from the original 48x48 which is a process that must be done for both the VGG-16 and ResNet50 models.

A huge benefit to this architecture is the rapid training time as shown in *Figure 6*. This allowed us to experiment more with this architecture than with others. As such this model was trained using both the raw fer2013 data, data that passed through the face extraction pre-processing with OpenCV, and the augmented data.

❖ *ResNet50 CNN Architecture*

ResNet50 is a convolutional neural network with 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer introduced in 2015 by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun [3]. It has been proven to have great prediction performance on facial data, with a standard and stable CNN structure [5].

We began the training process by reproducing result from *Facial Expression Recognition with Deep Learning* paper [2], which achieved the highest test accuracy at 73.2% in FER2013 challenges using ResNet50. Like the data processing procedure in VGG16, we resize and recolor the raw data (48, 48, 1) to match the input requirement and maximum training efficiency (197, 197, 3). To fine-tune the model, we froze all convolutional layers in ResNet except the last five layers and placed the original output layers with two FC layers sizes 4096 and 1024 and a 7-class SoftMax output layer.

After exploring different hyperparameters, we decided to use Adam optimizer with our best learning rate of 0.001 and a batch size of 128. Even though we applied several regularization methods including early stopping and dropout, our model is still slightly overfitted to training data due to small sample size and model complexity. (*Figure 5*)

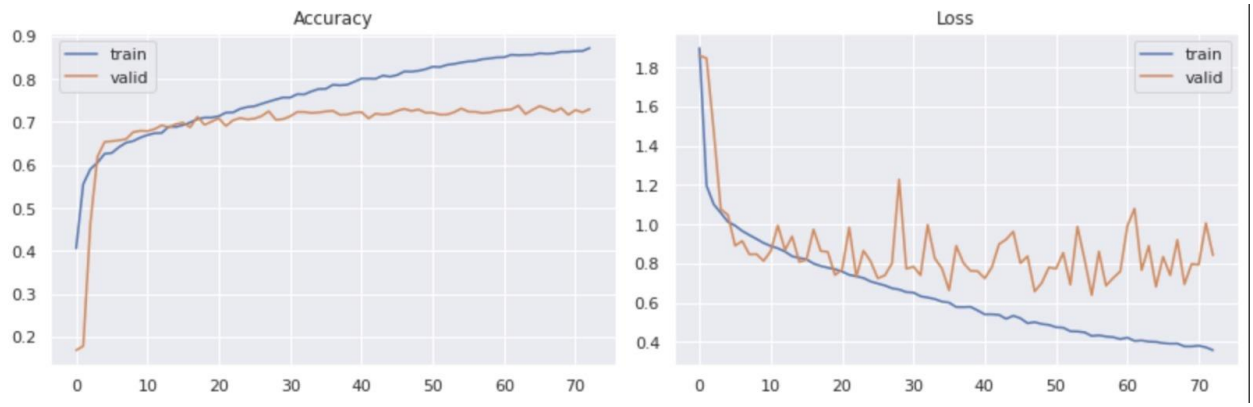

Figure 5. Accuracy and loss curves over epochs (Adam Optimizer)

❖ *Model evaluation*

The machine learning problem in this context is a multi-class classification task with seven different emotions. Specifically, seven labels are imbalanced in the fer2013 dataset. Compared to the remaining five emotions, there is a large proportion of happy faces and a small proportion of disgusted faces. Therefore, to examine the predictive performance for all emotions, we evaluated and selected the best model based on the overall accuracy, confusion matrix, and ROC on the testing set of the fer2013 dataset.

Since we are interested in real-time facial expression recognition in the zoom conference setting, we evaluated the model predictive accuracy on screenshots of the zoom participant pages and recorded the time elapsed for prediction. To collect these testing data, we recruited volunteers and asked them to make certain facial expressions according to the script describing different zoom conference scenarios. Finally, we compared the model results with human performance (data gathered from literature) on predictive accuracy and time efficiency to address the practicality of our application.

Modeling Results

The human baseline found in literature is shown to be 65% accuracy, according to research done by Ian J.Goodfellow's group in 2015 [13]. Recall that this is not an easy task - as mentioned in the *Data* section, some images are very subtle in expression or very similar across classes to be easily classified by human participants. Therefore, more complex neural networks are needed to detect the more subtle differences in expressions. In practice, VGG16 yields a slightly worse accuracy than the human baseline with 62%. This is followed by the CNN 10+ model which yields accuracies between 62-65% depending on the type of data preprocessing done. The only model to clearly outperform the human baseline is ResNet50 with an accuracy of 72%. A full comparison table of the models as well as the performance evaluation graphs for the ResNet50 model are shown below.

Model	Parameter	Training Time	Test Accuracy
(Human Level)	-	-	65% ± 5%
VGG-16 (Baseline)	94,418,759	4.7 hrs (256s/epoch)	62%
CNN 10 +	5,905,863	6.5 min	62%
CNN 10+ w/ OpenCV	5,905,863	5.75 min	64%
CNN 10+ w/ Data Augmentation	5,905,863	0.5 hrs (20 s/epoch)	65%
ResNet-50	96,699,520	5.2 hrs (256s/epoch)	72%

Figure 6. Comparison table for the various CNN architectures evaluated showing the number of parameters in the model, the total training time, and the final test accuracy.

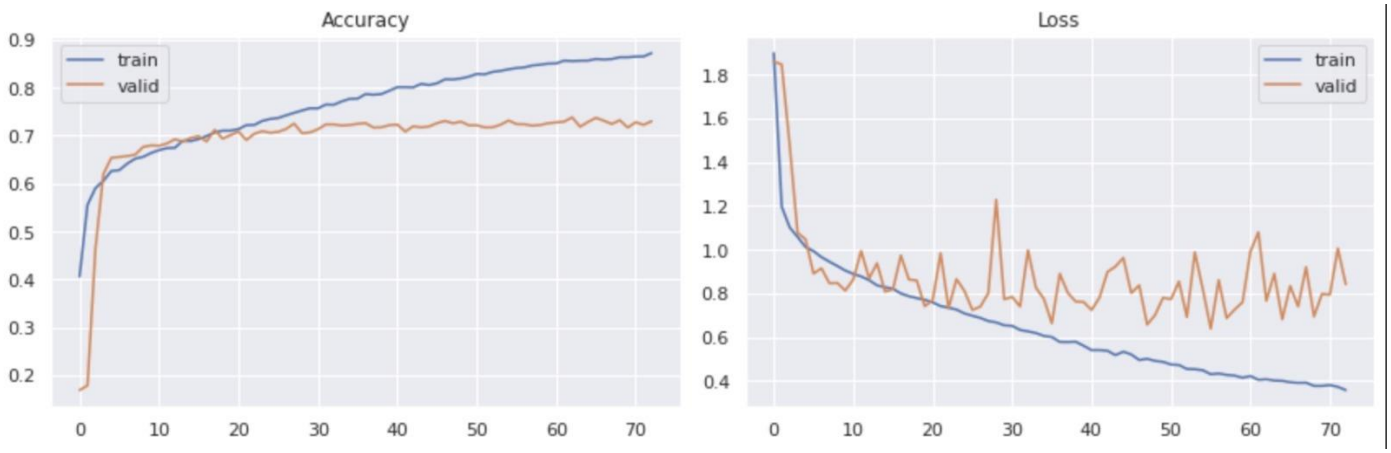

Figure 7. Accuracy and Loss curves for the ResNet50 model.

❖ *ROC & Confusion Matrix on ResNet50*

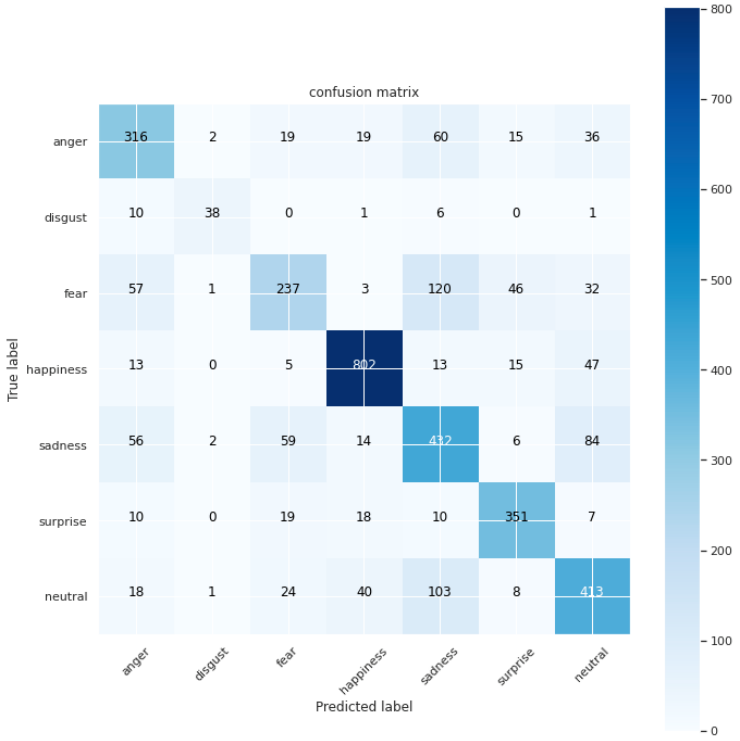

Figure 8. Confusion matrix for the ResNet50 model.

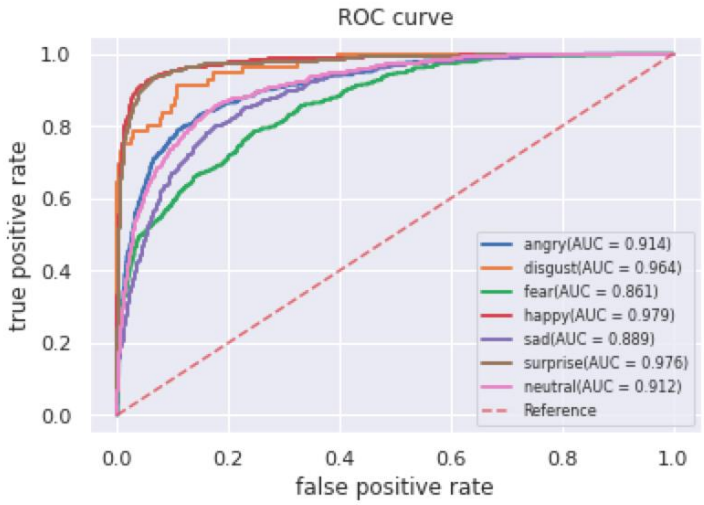

Figure 9. ROC curves for the ResNet50 model. Individual curves are shown for each class.

As shown in *Figure 9*, the model performs well in classifying happy and surprise faces, with overall AUC score of 0.97. As shown in *Figure 8*, 89% of faces predicted as happy are correctly labeled (precision) and 90% of happy faces are correctly identified (recall).

Investigating the cells in *Figure 8*, we found sadness and fear have relatively high misprediction rates. Only 58% of the predicted sad faces are sad, while 38% of them belong to fear and 33% are neutral. For fear

class, we observed a low recall rate at 48%, indicating more than half of the fear faces are predicted as sadness, fear, and other emotions.

This finding is interesting because fear and sadness rarely appear in remote learning settings [14] and even hard to distinguish and interpret by human. As we believe, in our Zoom application, measuring and providing meaningful feedback of the room is more important than label each negative emotion correctly, we will evaluate the model in a simulated zoom setting. We also experimented an alternative way to solve this problem by incorporating time series and combining some negative emotions labels for better interpretability [14], but further exploration on OpenCV algorithm is needed. Moreover, measuring the overall dynamic of the class is always more important than distinguishing between negative emotions in our application. Therefore, we also tested our model in simulated zoom conference condition to evaluate the performance.

- **Evaluation on Zoom Participant pages**

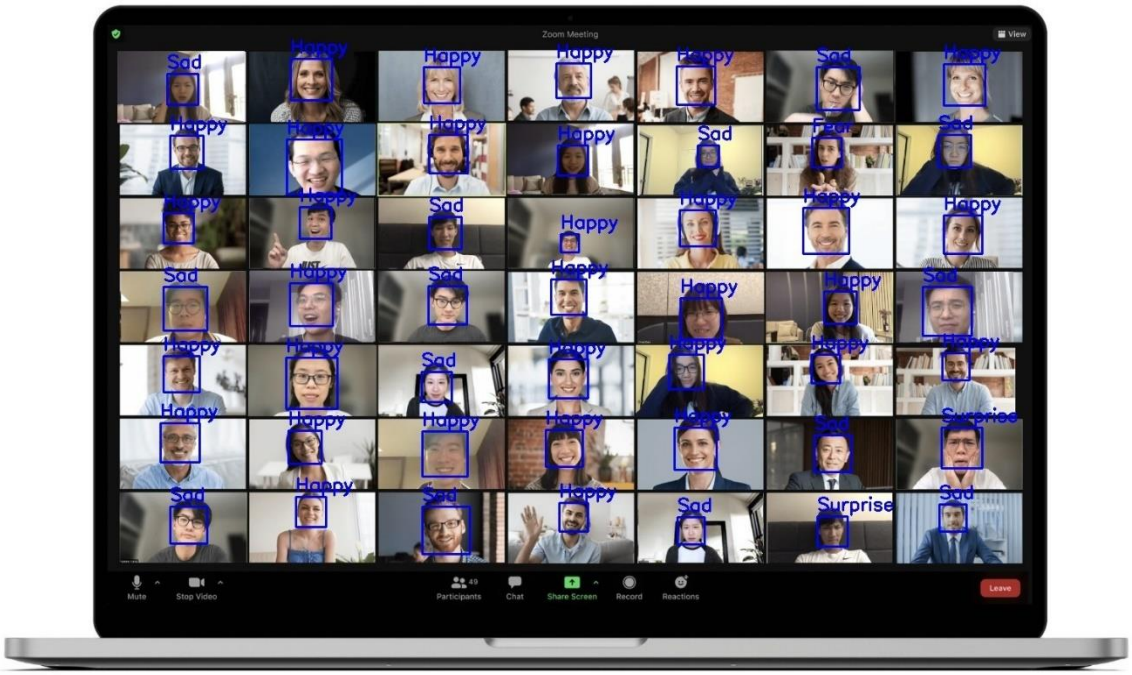

Figure 10. Sample video conference with predicted participant emotions.

We evaluated predictive accuracy and time efficiency of our application on the 7x7 Zoom gallery view page (*Figure 10*). Generally, the application took 0.13 seconds to make a prediction for a single participant. For a 7x7 zoom gallery view, the application successfully detected all faces and spent 5.04 seconds to generated predictions for the 49 participants in 5.04 seconds. Within a limited time, such as 5 seconds, humans tend to focus on a portion of faces and other faces fall into peripheral vision, which is weak in detecting details. From our own experiment, it would take about 30 to 35 seconds for humans to obtain a concrete understanding in the facial sentiment for a 49- participants meeting. Therefore, this application scales up well for multiple participants. One aspect of any future work should be to generate a concrete metric for comparing human performance to application performance on this task.

For the predictive accuracy of emotions, we found that the application is good at detecting happy and surprised faces while bad at detecting neutral and disgusted faces, which corresponds to our results on the

confusion matrix and ROCs. It is a reasonable finding because there are more training data for happy faces while significantly fewer training data for disgusted faces.

In practice, we realized that many neutral faces are misclassified as sad or fearful faces. This problem drew our attention because in a video conference setting, most participants would have neutral faces during the meeting and occasionally react to the speakers. According to the confusion matrix, our application does not perform well on detecting neutral faces in the fer2013 dataset. To improve the predictive accuracy on neutral faces, we might need more training data for the model.

In addition, we trained the model on the fer2013 dataset and tested on the faces cropped using OpenCV. From *Figure 10* above we can see that some cropped faces are do not include chins, which could potentially result in lost information that could assist in making an accurate prediction. The OpenCV face detection algorithm also requires an unobstructed view of a face to detect it. Humans, however, recognize emotions based on not only the faces but also the gestures/postures. For example, if one person frowns and covers the mouth with one hand, humans can predict this person is not happy. This type of information is lost when observing a static image that requires a clear view of the face.

Simply put, in the real-world setting, humans would likely perform better than this application in terms of predictive accuracy. This advantage likely fades as the number of faces to observe increases.

Conclusions

In general, this paper tackles two problems using open-source packages such as OpenCV and existing deep learning frameworks. The first problem is training a deep learning model that detects facial recognition and emotion with acceptable accuracy. Popular simple CNN, VGG16, and Resnet50 models are examined and trained. One major challenge is reshaping input size into three and four dimensions for VGG16 and ResNet50 respectively. The other one is data augmentation that requires creating separate directories for labels and images. The best transfer learning model is ResNet50 with 0.72 accuracy, which is 0.1 higher than the baseline model, VGG16.

Meanwhile, a real-world application is built to model Zoom settings. To model a real-life scenario, emotions such as fear and disgust are hardly appearing and simply collapsed into a broader “sad” category. OpenCV is utilized to capture video feeds in a given time period and return labeled sentiment near the faces of participants. The application works better in classification speed and recognizing faces at the edge of the Zoom gallery view, which is usually ignored compared to faces at the center by human eyes. Detailed instructions on recreating this research can be found in our Github repository.

This application can benefit professors in knowing how the entire class feels about the lecture content by calculating the percentage of different faces. Taking faces at the edge into consideration, the application is more likely to avoid bias incurred from only looking at the center of the Zoom gallery view. Future improvements mainly include two aspects, implementing distraction-free algorithms and publishing a complete website application. As for distraction, an algorithm can be implemented before emotion detection to ensure that only faces of people who turn their cameras on and are looking at the camera are used afterwards. Those who are “distracted” will also be marked to the professor. With regards to difficulty in recreation, a complete website application may also be built or containerized so that users can access the application with a click instead of going through the command line operations.

GitHub Link: https://github.com/yichenghuang980/emotion-detection

Roles

Han
- Trained and evaluated CNN 10+ w/ data augmentation
- Implemented code for VGG16 & ResNet50
- Contributed to method and result section
- Video lead: edited videos and generated experiment condition

Zoe
- Trained and evaluated the baseline CNN architecture
- Designed and analyzed real-word test for the application
- Contributed to model evaluation and the corresponding result section
- Proposal and progress report lead

Tommy
- Trained one of the VGG16 model without OpenCV
- Reviewed and found papers for aggregated evaluation metric
- Final report writing (abstract, introduction, background)
- Managed the timeline of the project progress

Wilson
- Drafted conclusion and part of methods section
- Github Lead: update README and command line code for recreation
- Trained one of the VGG16 model with OpenCV
- Toyed with Resnet50 model

Malcolm
- Wrote all code for the application
- Trained and evaluated CNN 10+ w/ and w/o OpenCV
- Final report formatting and proofing

Timeline of activity

- ❖ Brainstorm project ideas - Feb 15, 2021
  - Discuss topics/questions of interest
    - Facial expression recognition
    - Food recognition
    - Video processing
- ❖ Explore project ideas & Draft proposal - Feb 20, 2021
  - Discuss the real-world application
    - Emotion/engagement detection for online communication(eg Zoom lectures)
  - Determine the machine learning component in this project
  - Propose plan B: Recommendation system for Instacart reorder
  - Assigned group roles and started literature review
- ❖ Check with instructor - Feb 22, 2021
  - Receive feedback from instructor on project ideas and ML methods
    - Explore multitask learning?
    - Design a group sentiment metric?

        - Add visualization for engagement?
    - Finalize the project proposal
- ❖ Reflect on the proposal feedback – Mar 5, 2021
    - Build and evaluate the model on static images rather than videos
    - Consider measure computational efficiency for real-time prediction
    - OpenCV + CNN architecture, discussed potential applications
    - Decide model evaluation strategies
- ❖ Progress meeting – Mar 12/14, 2021
    - Explore OpenCV & Potential CNN architectures
    - Discuss definition & evaluation for group sentiment & engagement
- ❖ Complete progress report – Mar 20/23, 2021
- ❖ Progress meeting - Mar 27, 2021
    - Literature review & exploration on Transfer Learning & Data Augmentation
    - Integrate with OpenCV
- ❖ Reflect on the progress report feedback – Apr 13, 2021
    - Improve the model, Train ResNet
    - Design real-world test for the application
- ❖ Complete the presentation – Apr 18, 2021
- ❖ Complete the final report – Apr 24, 2021

References

[1] Li, B., & Lima, D. (2021). Facial expression recognition via resnet-50. *International Journal of Cognitive Computing in Engineering, 2*, 57-64. doi:10.1016/j.ijcce.2021.02.002

[2] Dhankhar, P. (2019). ResNet-50 and VGG-16 for recognizing Facial Emotions. *International Journal of Innovations in Engineering and Technology (IJIET), 13*(4), 126-130. http://ijiet.com/wp-content/uploads/2019/08/18.pdf

[3] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2016.90

[4] Simonyan, K., & Zisserman, A. (2015, April 10). Very deep convolutional networks for large-scale image recognition. Retrieved April 24, 2021, from https://arxiv.org/abs/1409.1556v6

[5] Khanzada, A., Bai, C., & Celepcikay, F. T. (2020). Deep learning approach for facial expression recognition. *Journal of Critical Reviews, 7*(14). doi:10.31838/jcr.07.14.108

[6] Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014, November 06). How transferable are features in deep neural networks? Retrieved April 24, 2021, from https://arxiv.org/abs/1411.1792v1

[7] Sharma, P., Joshi, S., Gautam, S., Maharjan, S., Filipe, V., & Reis, M. (2020, December 26). Student engagement detection using emotion Analysis, eye tracking and head movement with machine learning. Retrieved April 25, 2021, from https://arxiv.org/abs/1909.12913

[8] Parkhi, O. M., Zisserman, A., & Vedaldi, A. (n.d.). VGG face descriptor. Retrieved April 25, 2021, from https://www.robots.ox.ac.uk/~vgg/software/vgg_face/

[9] AlMarri, S. B. S. (2019). Real-Time Facial Emotion Recognition Using Fast R-CNN. https://scholarworks.rit.edu/cgi/viewcontent.cgi?article=11364&context=theses

[10] Gitshanks. Gitshanks/fer2013. Retrieved April 25, 2021, from https://github.com/gitshanks/fer2013

[11] Tang, Travis. Travistangvh/emotion-detection-in-real-time. Retrieved April 25, 2021, from https://github.com/travistangvh/emotion-detection-in-real-time

[12] Atulapra. Atulapra/Emotion-detection. Retrieved April 25, 2021, from https://github.com/atulapra/Emotion-detection

[13] Goodfellow, I., Erhan, D., Carrier, P., Courville, A., Mirza, M., Hamner, B., . . . Bengio, Y. (2014, December 29). Challenges in representation Learning: A report on three machine LEARNING CONTESTS. Retrieved April 25, 2021, from https://www.sciencedirect.com/science/article/pii/S0893608014002159

[14] Mukhopadhyay, M., Pal, S., Nayyar, A., Pramanik, P. K. D., Dasgupta, N., & Choudhury, P. (2020, February). Facial Emotion Detection to Assess Learner's State of Mind in an Online Learning System. In Proceedings of the 2020 5th International Conference on Intelligent Information Technology (pp. 107-115). https://dl.acm.org/doi/abs/10.1145/3385209.3385231

[15] Krithika L.B, & Lakshmi Priya GG. (2016). Student emotion recognition System (sers) for E-LEARNING improvement based ON LEARNER Concentration Metric. Procedia Computer Science, 85, 767-776. doi:10.1016/j.procs.2016.05.264

[16] Bouhlal, M., Aarika, K., Abdelouahid, R. A., Elfilali, S., & Benlahmar, E. (2020). Emotions recognition as innovative tool for improving students' performance and learning approaches. Procedia Computer Science, 175, 597-602. doi:10.1016/j.procs.2020.07.086

[17] Nayak, S. (2021, April 21). Image classification using transfer learning in pytorch: Learn opencv. Retrieved April 26, 2021, from https://learnopencv.com/image-classification-using-transfer-learning-in-pytorch/

[18] Jennifer, J. S., & Sharmila, T. S. (2017). Edge based eye-blink detection for computer vision syndrome. 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP). doi:10.1109/icccsp.2017.7944084

[19] Utami, P., Hartanto, R., & Soesanti, I. (2020, January 02). A study on facial expression recognition in assessing teaching skills: Datasets and methods. Retrieved April 27, 2021, from https://www.sciencedirect.com/science/article/pii/S1877050919318654