CS273A Project Report: Emotion Detection dataset

Group 11: Team Navarasa

Gnana Heemmanshuu Dasari - ghdasari@uci.edu

Sumit Chandrashekhar Raut - scraut@uci.edu

Sanjita Venkatesh Nayak - nayaksv@uci.edu

CS273A: Intro to Machine Learning

Prof. Alexander Ihler

December 12, 2024

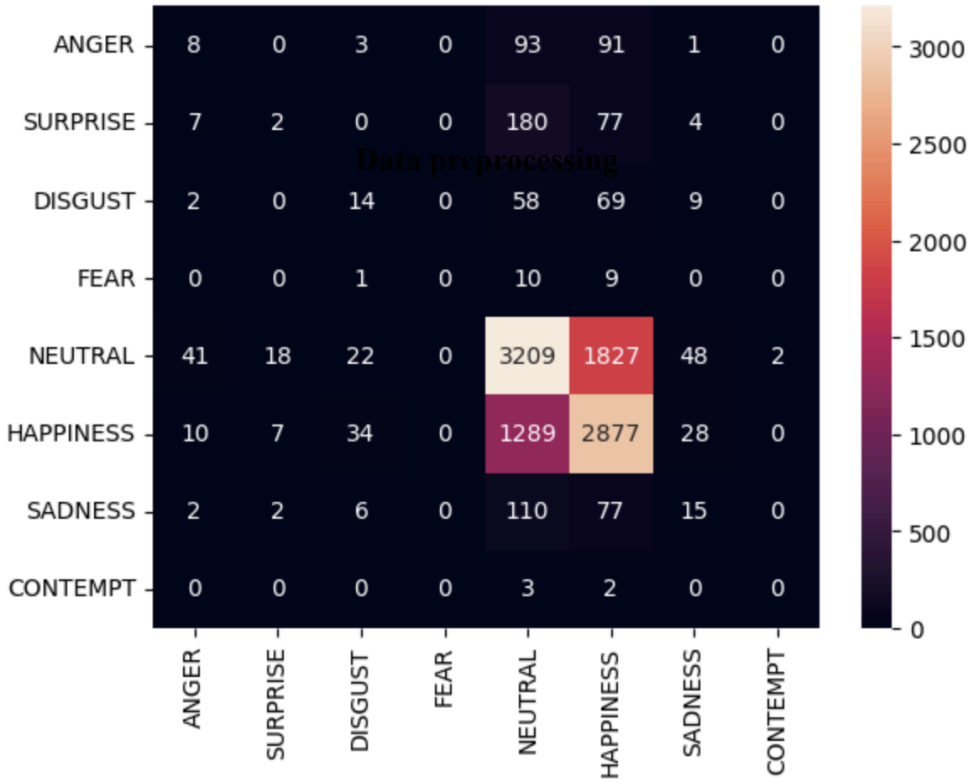
Dataset

The dataset we have analyzed is the Emotion Detection dataset here. It has 13,690 images spread across **8** emotions - ANGER, SURPRISE, DISGUST, FEAR, NEUTRAL, HAPPINESS, SADNESS, CONTEMPT. A legend.csv file contains the names of images mapped to the corresponding class, and the images are organized in a folder in JPEG format. The dataset is by no means standardised, the images are of various different shapes. Some are grayscale and some are colored. The occasional image is an off handed shape of (536, 355, 3), and hundreds such unique shapes exist. The most common shape of images in the dataset is (350, 350) ie., the images are grayscale. In addition, the class assignments in the data also vary, with some images having the class written in lowercase and some in uppercase. Such errors are bound to happen in a dataset which is open and free for contribution, without an explicit set of rules for contributors. The readme.MD of the dataset seems to be abandoned or still under works, as it does not state the set of emotions the dataset has, only specifying vague rules on image size which have also not been followed. The image organisation and naming convention in photos is also all over the place, but this is not a significant issue. We observed several other issues with the dataset that we have attempted to address.

Preliminary Approach

Given the success of Convolutional Neural Networks in image classification, we set out to construct a CNN that would do well on this data and compare it with other supervised learning methods such as K-nearest neighbors. In an effort to keep it simple, our first model had 1 Conv2D layer and 1 pooling layer followed by ReLU activation and softmax. We found its

performance through a confusion matrix on the training data indicating the biggest problem with our dataset:



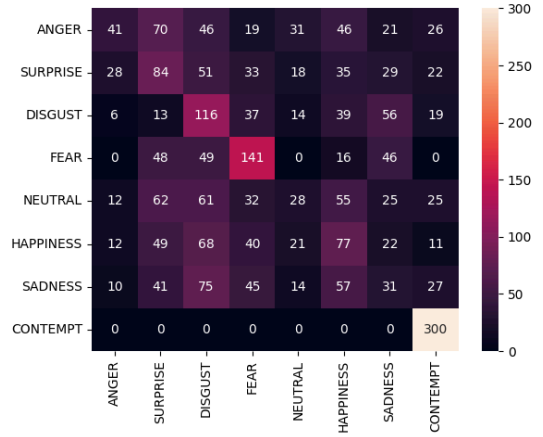
There is a lot of data solely focussed on the HAPPINESS and NEUTRAL classes, which we confirmed by going back to the data processing steps and checking the distribution of the full dataset (see right). As is evident from the confusion matrix, the data for other classes (especially FEAR and CONTEMPT) is extremely low, and these classes are rarely ever predicted by the model.

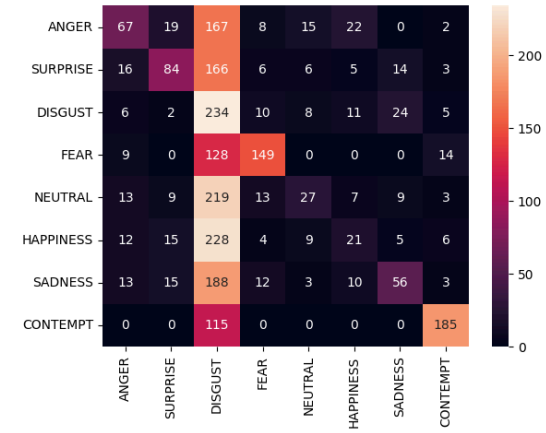
```
0, ANGER, 252
1, SURPRISE, 368
2, DISGUST, 208
3, FEAR, 21
4, NEUTRAL, 6868
5, HAPPINESS, 5696
6, SADNESS, 268
7, CONTEMPT, 9
```

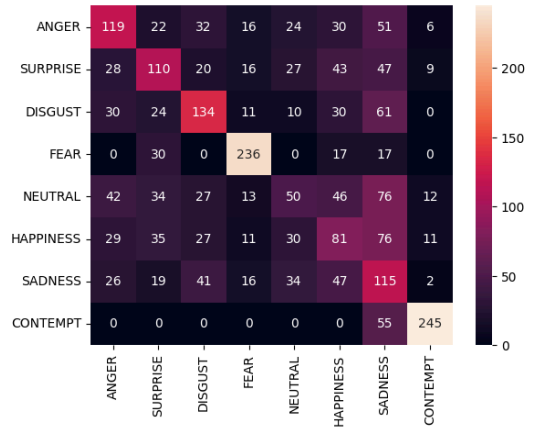
Addressing imbalanced data through sampling

With the dataset that we have, accuracy cannot be used as a success metric. The model can be biased towards predicting NEUTRAL/HAPPINESS and will still do well because most of the

data belongs to these classes. The model wouldn't be "learning" anything. Examination of the confusion matrix would be a better idea to gauge performance here so we stick to that. Being restricted to an exploratory analysis of this dataset, we attempt to find ways to make do with what we have. We started with a combination of random undersampling of images from the majority class and oversampling from the minority classes.





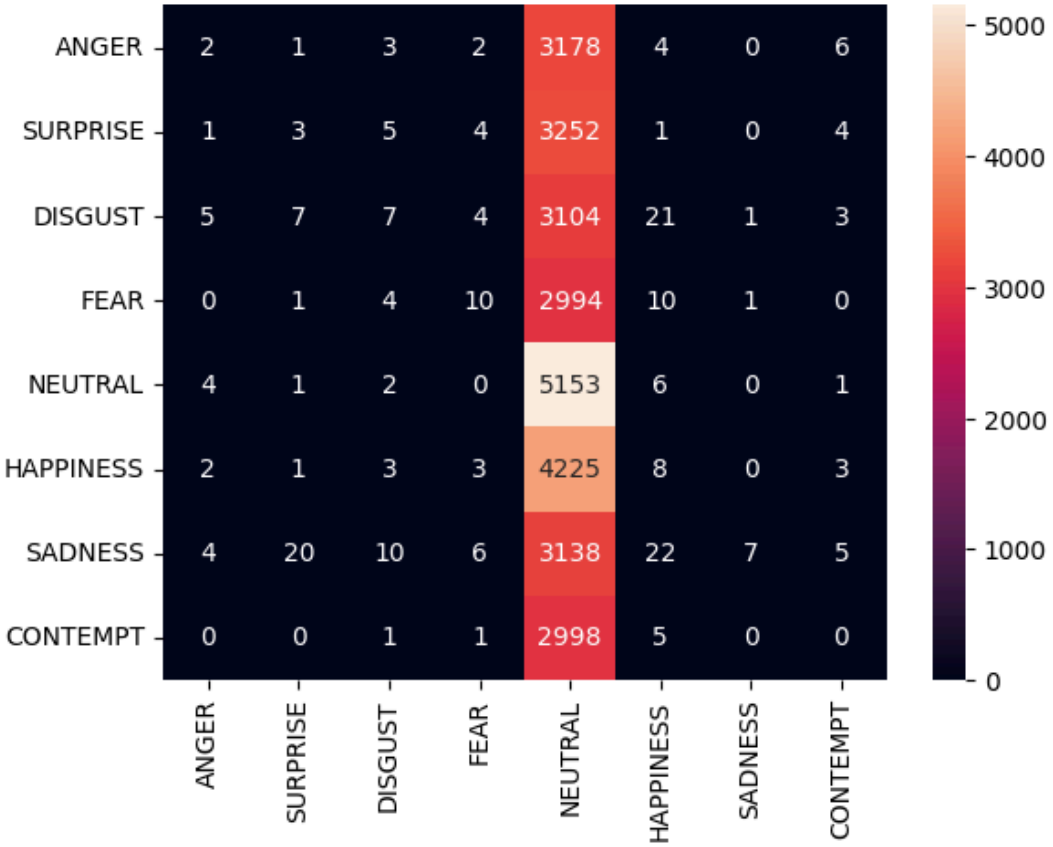


In each run, we get significantly different results. The target size for sampling was set to 300 for all classes to maintain uniformity. The problem with direct random sampling is the propensity for the loss of information caused by losing out on a chunk of the majority data, combined with the propensity to overfit on the minority data because the data repeats itself so many times. Overfitting on the minority class labels can explain the huge variance seen in the training data confusion matrix above. Particularly in the case of the CONTEMPT class of which the original data has only 9 instances, this overfitting is evident.

Data Augmentation

Seeing the problems with oversampling and undersampling of data, we moved on to attempt balancing the dataset through data augmentation via simple transformations. On inspection we

realized that other techniques such as SMOTE (Synthetic Minority Over-sampling Technique) would not be effective here in image datasets. SMOTE operates by assuming and generating new features by interpolating between existing samples, which doesn't account for the spatial structure of images. This can lead to unrealistic, blurry images and overfitting, especially when the minority class has very few samples, as SMOTE would struggle to create meaningful diversity in such cases. We tried an augmentation approach to increase the number of images of each class to 3000. The augmentation is more pronounced for classes like FEAR and CONTEMPT which have very few images, and somewhat moderate for the other classes. Classes NEUTRAL and HAPPINESS were not augmented at all. Contrary to expectation, however, there is a significant decrease in loss (90%), but the confusion matrix shows that the model is just predicting NEUTRAL for all classes. This is also evident by the loss barely improving from 0.14 to 0.16.

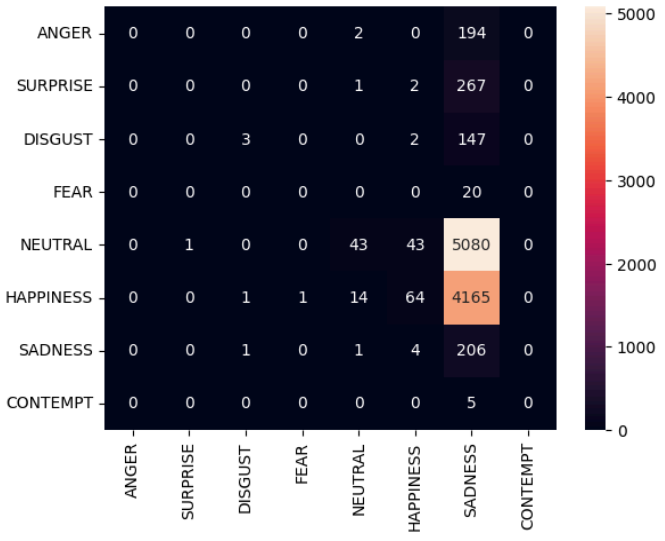


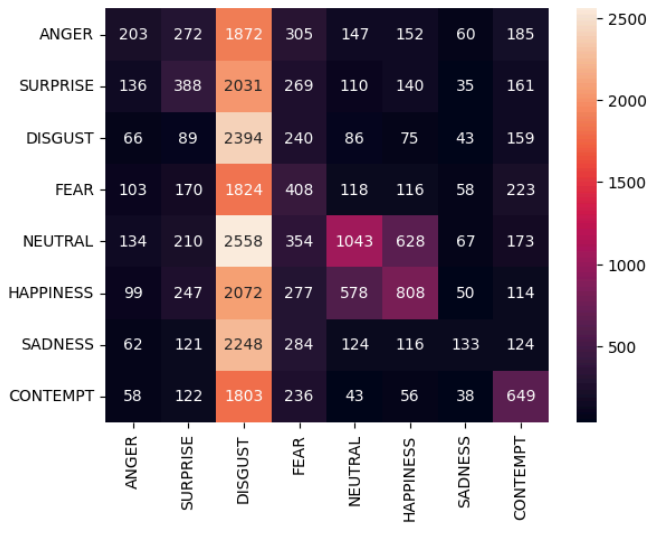
	precision	recall	f1-score	support
ANGER	0.43	0.02	0.03	196
SURPRISE	0.34	0.07	0.11	270
DISGUST	0.67	0.01	0.03	152
FEAR	0.00	0.00	0.00	20
NEUTRAL	0.59	0.33	0.42	5167
HAPPINESS	0.46	0.79	0.58	4245
SADNESS	0.31	0.02	0.04	212
CONTEMPT	0.00	0.00	0.00	5
accuracy			0.49	10267
macro avg	0.35	0.15	0.15	10267
weighted avg	0.52	0.49	0.46	10267

	precision	recall	f1-score	support
ANGER	0.23	0.00	0.00	3196
SURPRISE	0.33	0.00	0.00	3270
DISGUST	0.56	0.00	0.01	3152
FEAR	0.30	0.00	0.01	3020
NEUTRAL	0.18	1.00	0.31	5167
HAPPINESS	0.07	0.00	0.00	4245
SADNESS	0.46	0.01	0.02	3212
CONTEMPT	0.00	0.00	0.00	3005
accuracy			0.18	28267
macro avg	0.27	0.13	0.04	28267
weighted avg	0.25	0.18	0.06	28267

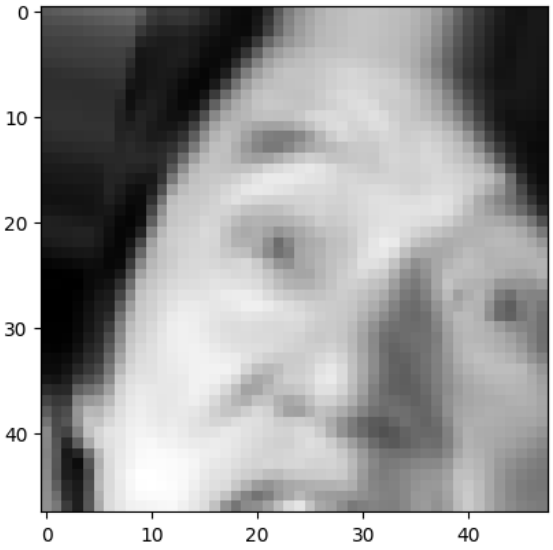
Looking at the f1 scores (before and after, respectively) gave us a better understanding of why this might have happened. The model has evidently overfit to the NEUTRAL class (indicated by the perfect recall 1.00 on the right). This might have been due to the augmentation (which was performed on all classes except NEUTRAL and HAPPINESS has led to class noise in the minority classes. The minority classes were already extremely small. The augmentations generated from these small classes may have been insufficiently diverse or overly similar, failing to provide the model with meaningful variations.

As a final measure, we added weights to each class in the model, both before and after performing augmentation to see how the performance would change. The class weights computed using sklearn were mostly close to 1 in the augmentation case, but were vastly different in the very initial model we had, ranging from 0.3-250.





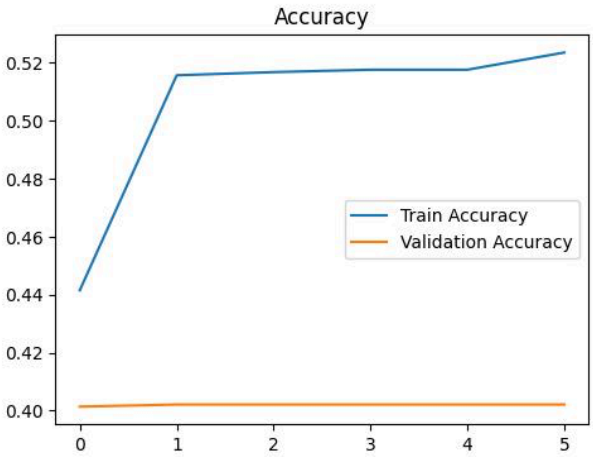
As seen on the left, the model overfits onto different classes each time it is trained with computed class weights, while a similar overfit is present in the model even after augmentation. Such behaviour was consistently observed with varying levels of augmentation in the data.

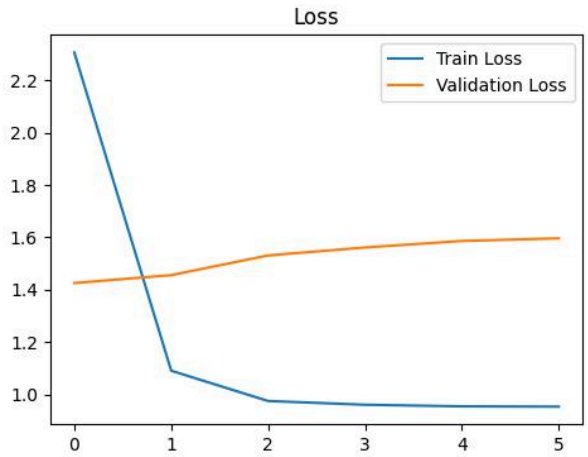


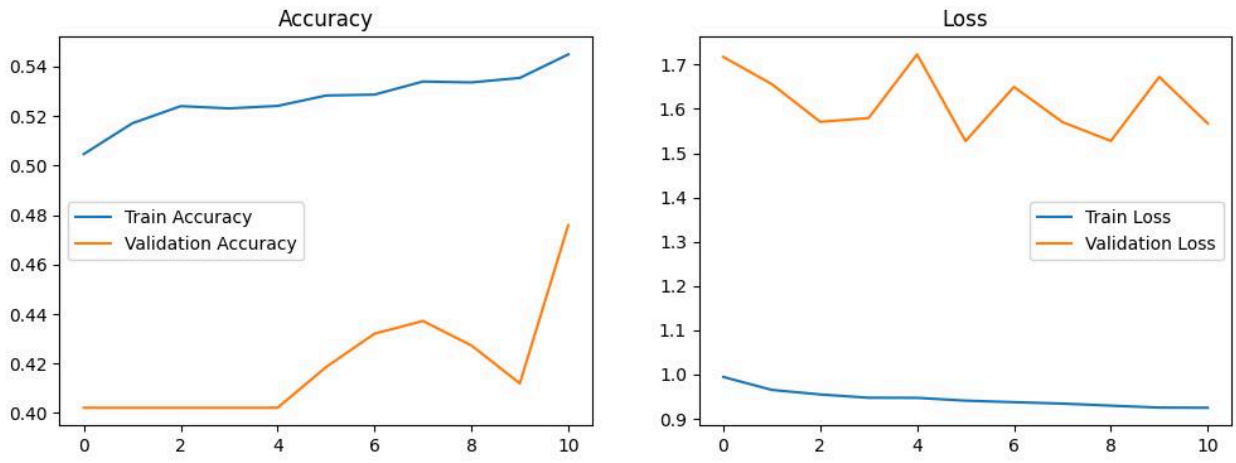
A sample augmented image. It is possible that due to the original dataset having images that mostly fit the frame, the augmented data is not very realistic with unnatural distortions.

Fine Tuning the CNN

Separately, we also added two more Conv2D layers to our model, with Pooling layers after each and a couple of Dropout layers. We also compared our model with the k-Nearest Neighbor model. The kNN model performed best at k=101, so this model was used for the final comparison. The accuracy and loss of the models are as follows. The finer layer details and parameter values can be found in the code:







Our final Convolutional Neural Network model

Discussions and Future Work

Having worked on this rather challenging dataset, we are further motivated to work on other emotion detection related datasets that are perhaps more balanced, to enable focus on parameter tuning for CNNs. We attempted to use pre-trained models such as ResNet but were unable to, as ResNet is trained on and seems to work only with RGB images with 3 channels. One unexplored idea with this dataset is image synthesis, not through techniques like SMOTE but through Generative Adversarial Networks (GANs) or other advanced augmentation techniques, to generate realistic minority class samples and improve model performance. We also plan on exploring one idea that we did not have enough time to, that of optimising the model on f1-score instead of traditional loss functions. It cannot be optimised directly with gradient descent, but a "softer" version using probabilities instead of true and false positives can be used. We plan to evaluate how this approach fares on our dataset.

References and acknowledgements

With this project essentially being one of our firsts in ML, we have referred to and taken inspiration from a wide number of sources on the internet, including source codes on Github, answers from StackOverflow, blogs written on the subject and ChatGPT - mostly to get an idea of why a problem might be observed and what are the ways we can address them. Each such address has been tested out and retained if improvements were observed, as discussed in the report. We have tried to list all of our major references below.

ProjectPro (2024), *Facial Emotion Recognition Project using CNN*. Retrieved Dec 12, 2023, from https://www.projectpro.io/article/facial-emotion-recognition-project

Khan, Zaid (2023), *Real-time Facial Emotion Recognition using Deep Learning and OpenCV.* Retrieved Dec 12, 2023 from https://fuyofulo.medium.com/real-time-facial-emotion-recognition

Balaji, Atul (atulapra), *Emotion-detection*. Source code from GitHub. https://github.com/atulapra/Emotion-detection

Computer vision engineer (YouTube, 2022), *Emotion detection with Python and OpenCV*. https://www.youtube.com/watch?v=Vq_01gFG2vk

Manish (manish-2945), *Facial-Emotion-Recognition-using-OpenCV-and-Deepface*. Source code from Github. https://github.com/manish-9245/Facial-Emotion-Recognition-using-Deepface

Kothiya, Y (2019), *How I handled imbalanced text data.* https://towardsdatascience.com/how-i-handled-imbalanced-text-data-ba9b757ab1d8

Alencar, R (2017), *Resampling strategies for imbalanced datasets*. https://www.kaggle.com/code/rafjaa/resampling-strategies-for-imbalanced-datasets

StackOverflow, *Dealing with class imbalance in a neural network.* https://datascience.stackexchange.com/questions/73684/how-to-deal-with-class-imbalance

StackOverflow, *Optimizing neural networks with F-scores.*

https://stackoverflow.com/questions/65318064/can-i-trainoptimize-on-f1-score-loss-with-pytorch

Lucas, T (2022), *How to Deal with Imbalanced Datasets in Computer Vision*

https://www.picsellia.com/post/improve-imbalanced-datasets-in-computer-vision

Gupta, S. (2022), *Addressing Data Imbalance in Image Classification: Techniques and Strategies*

https://medium.com/@shubhamgupta.3101994/addressing-data-imbalance-in-image-classification

Billington, A (2023), *Image Classification — Dealing with Imbalance in Datasets*

https://www.advancinganalytics.co.uk/blog/2023/2/2/image-classification-dealing-with-imbalance-in-datasets

Pastor-Pellicer, J. (2013). *F-Measure as the Error Function to Train Neural Networks*. In: Rojas, I., Joya, G., Gabestany, J. (eds) Advances in Computational Intelligence. IWANN 2013. Lecture Notes in Computer Science, vol 7902. Springer, Berlin, Heidelberg.

https://doi.org/10.1007/978-3-642-38679-4_37