CS464 Project
Facial Expression Recognition

Group 8

Kürşad Güzelkaya
Süleyman Gökhan Tekin
Mustafa Mert Türkmen
Bahadır Yüzlü
Hazal Buluş

Introduction

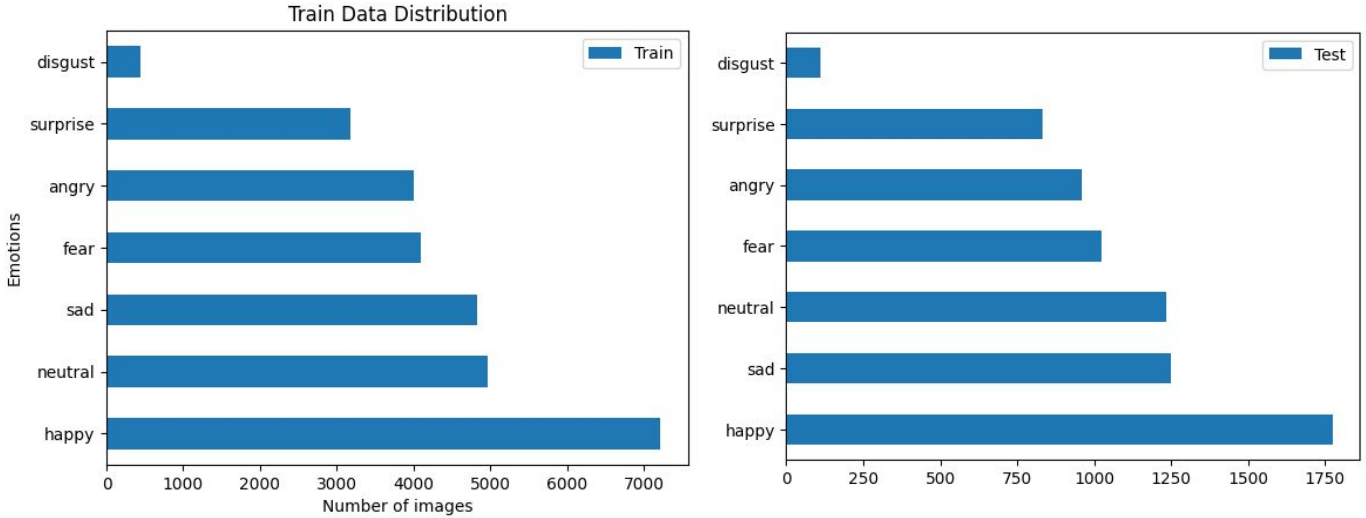
- Focuses on developing a facial expression recognition system
- Analyze and compare the performance of various machine learning and deep learning models:
  - K Nearest Neighbour (KNN), Convolutional Neural Network (CNN), Decision Trees, and Random Forest

Background Information of Dataset

- FER-2013
- Dataset consist of 48x48 pixel grayscale images of faces
- Each face is categorized based on the emotion
- Seven categories: 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral

Background Information of Dataset

- The training set consists of 28,709 examples, test set consists of 3,589 examples.
- Imbalanced dataset



Data Balancing Techniques

- Methods used to balance dataset:
  - Data Augmentation
  - Random OverSampling
  - Resampling
  - Synthetic Minority Oversampling Technique (SMOTE)
  - Balanced Bagging Classification

Data Balancing Techniques

Balanced Bagging Classification

- Additional Classification Procedure
- Combines bagging with balanced sampling.
- Ensures that each training subset has an equal mix of all classes.

Synthetic Minority Oversampling Technique (SMOTE)

- Aims to balance dataset by artificially generating new examples of the minority class.
- Creates synthetic examples that are combinations of the closest minority class cases.

Data Balancing Techniques

Resampling

- Resample arrays or sparse matrices in a consistent way
- Determine the class with the maximum samples (majority class)
- For each minority class:
  - Upsample the data to match the size of the majority class using resampling (with replacement).

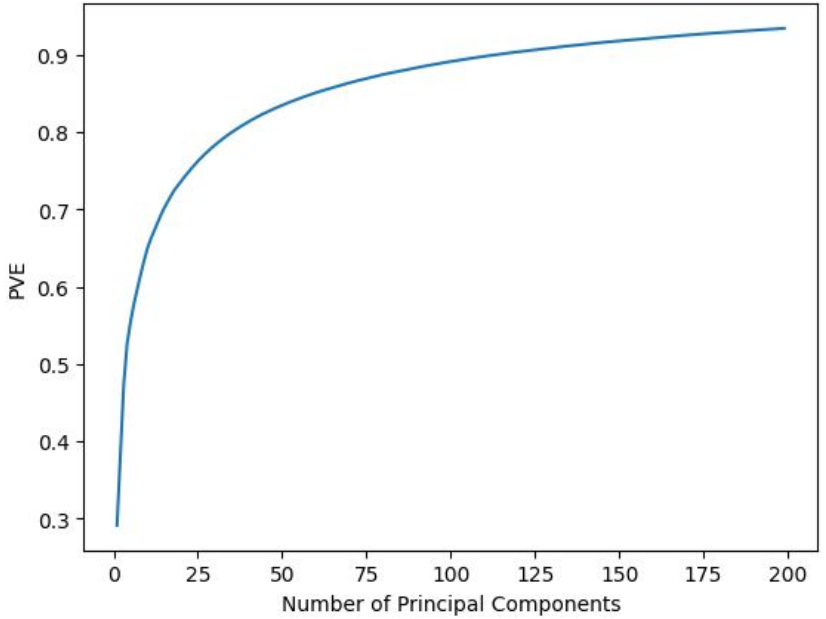
Random Over Sampling

- Increase the number of samples in minority classes by duplicating them.
- Match the size of the majority class to balance the dataset.
- Randomly select and replicate instances from minority classes.
- Often done with replacement for a balanced class distribution.

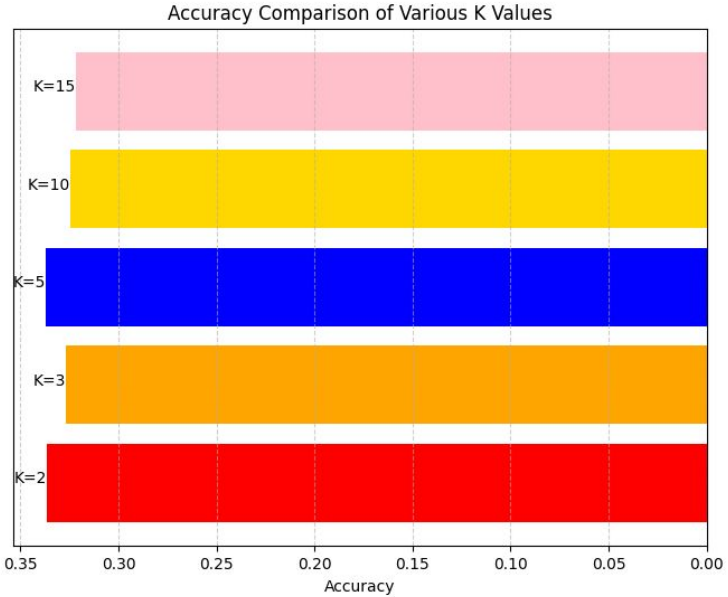
PCA- Explained Variance Analysis

- Most variance captured by the first 50 components.

- Plateau after 100 components, minimal variance gain.
- Adding more components does not significantly increase the explained variance.
- Helps in dimensionality reduction, retaining most of the information while simplifying the model



K Nearest Neighbour (KNN)



- Maximum accuracy on imbalance data with K=5.
- PCA applied with 100 and 200 components, K is fixed to 5.
- Cause of getting low accuracy with PCA may be class Imbalance, or loss of important information, or overfitting.

```
Accuracy with 5-NN on 100 principal components: 18.80%
Accuracy with 5-NN on 200 principal components: 18.47%
```

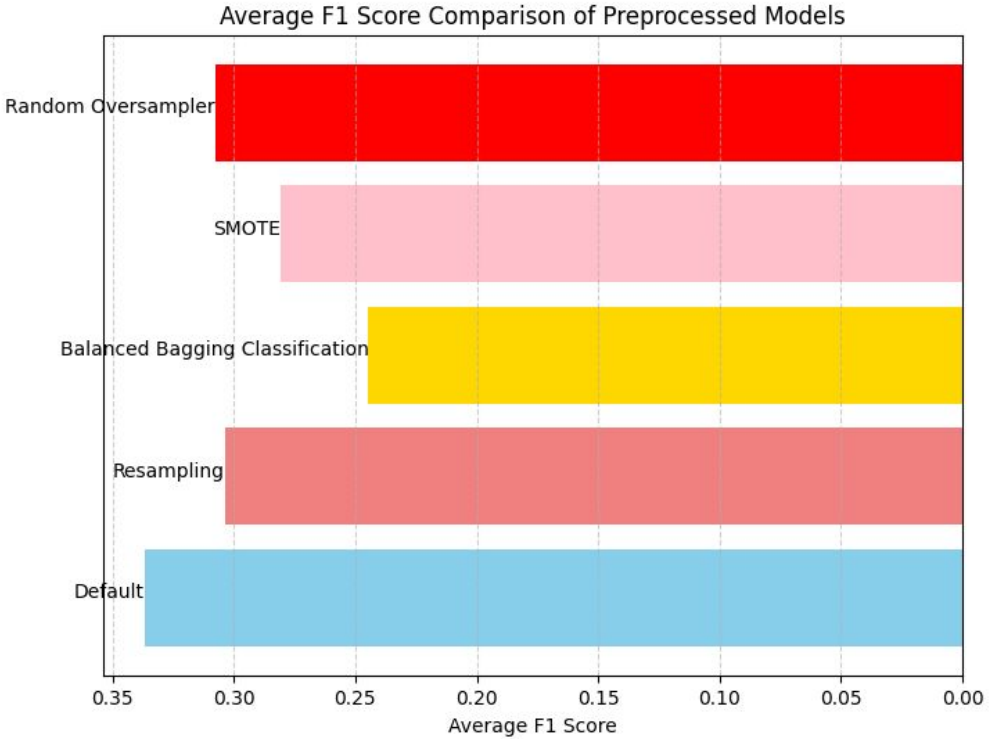
K Nearest Neighbour (KNN)

Preprocess Comparison

F1 performance dropped post-balancing.

Possible reasons:

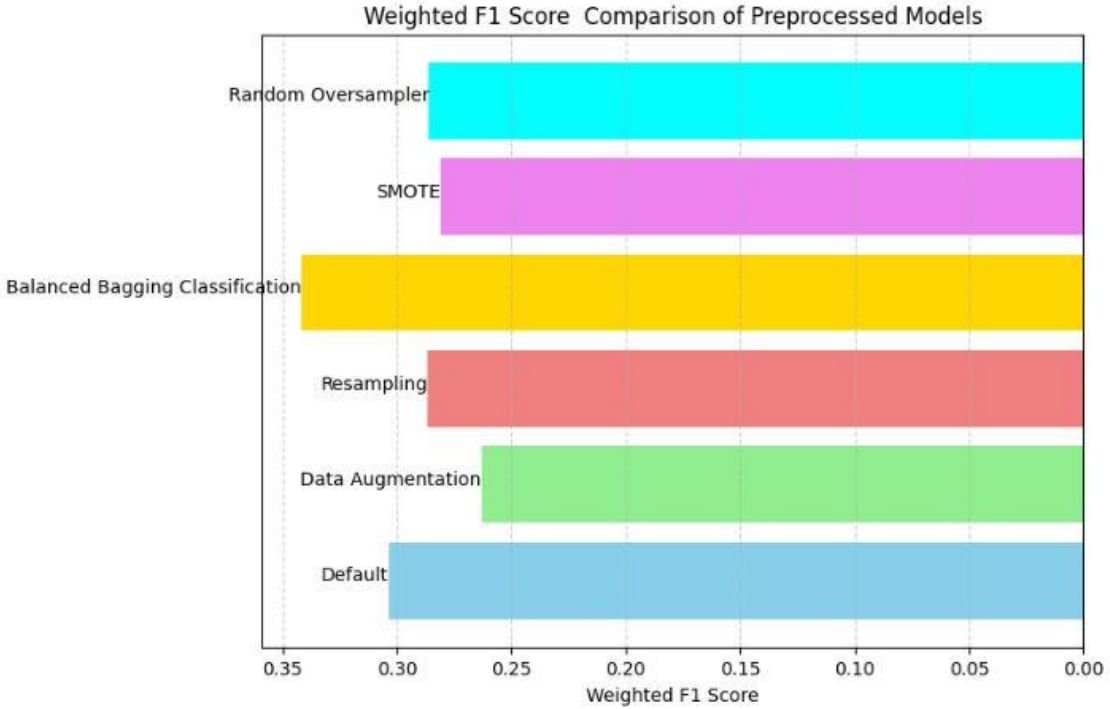
- Oversampling may cause the model to overfit the minority class, hurting generalization.
- Balancing can distort data relationships, impacting model accuracy.
- Model parameters tuned for imbalanced data may perform poorly on balanced data.



Decision Tree Classifier

Preprocess Comparison

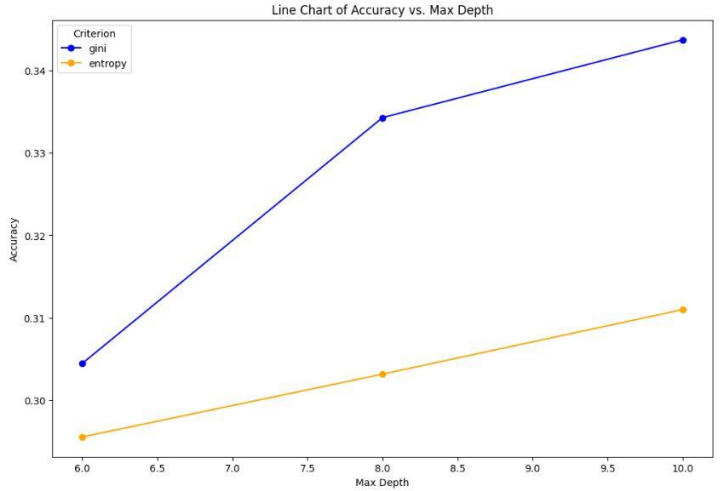
- Comparison of models
- F1 Score
- Result
- Flattening
- Not Scaling

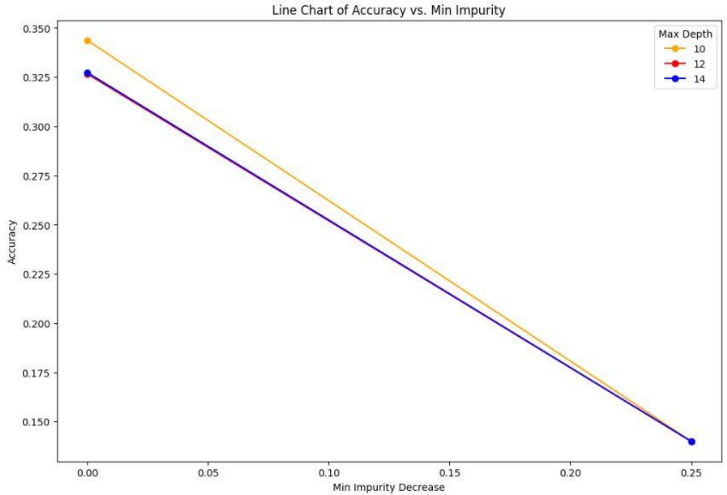


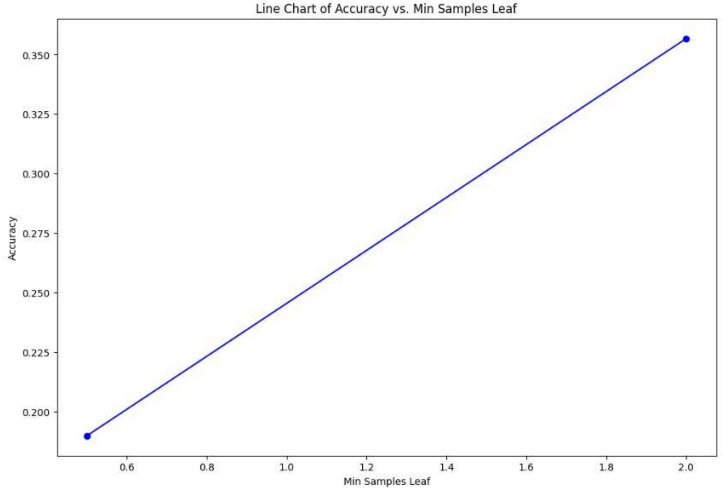
Decision Tree Classifier

Tuning

- Prevent Overfitting and Underfitting
- Grid Search
- Accuracy with Validation Data
- Forward Parameter Selection







Decision Tree Classifier

Final Model

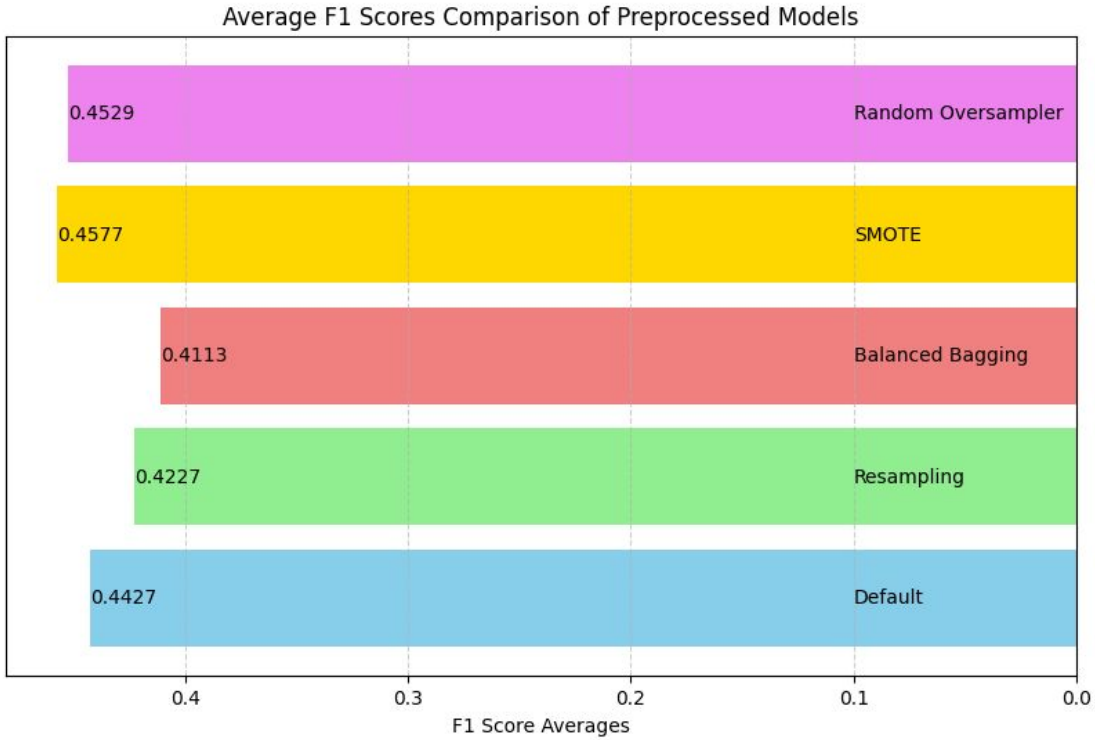
```
best_decision_clf=DecisionTr… = DecisionTreeClassifier(criterion = "gini",
                        random_state=42,
                        max_depth = 10,
                        min_impurity_decrease = 0,
                        min_samples_leaf = 2)

classifier_BBC_best=BalancedBa… = BalancedBaggingClassifier(base_estimator= best_decision_clf,
                                        n_estimators=50,
                                        sampling_strategy='auto',
                                        random_state=42)
```

- The best available model parameters
- Final Result on the Test Data

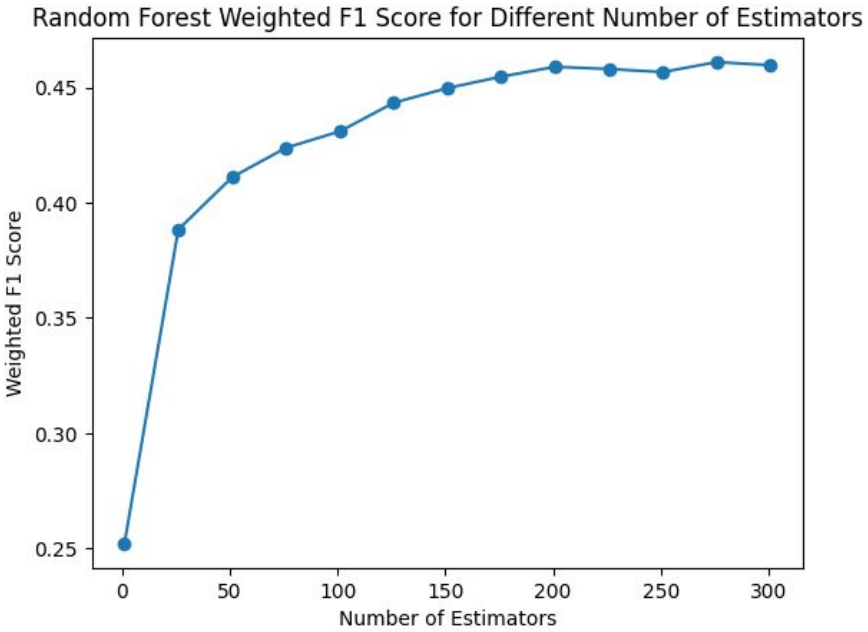
Random Forest Classifier

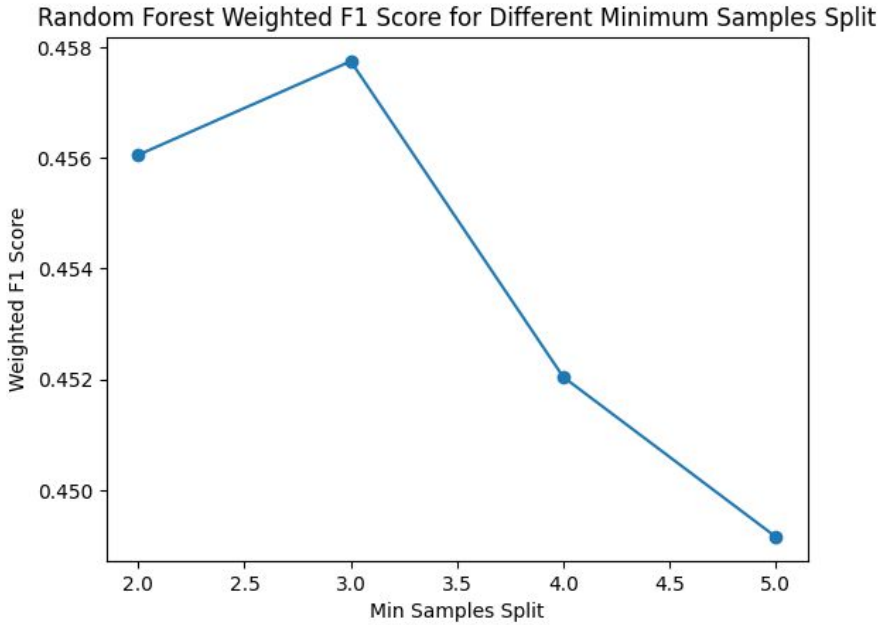
Preprocess Comparison



Random Forest Classifier

Tuning

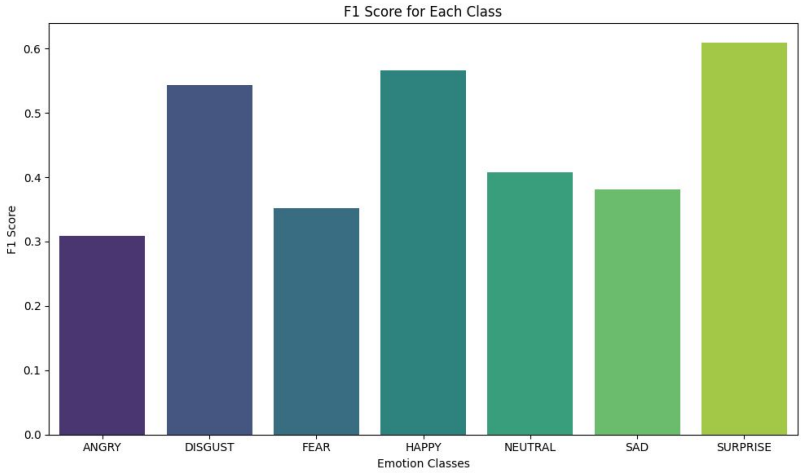


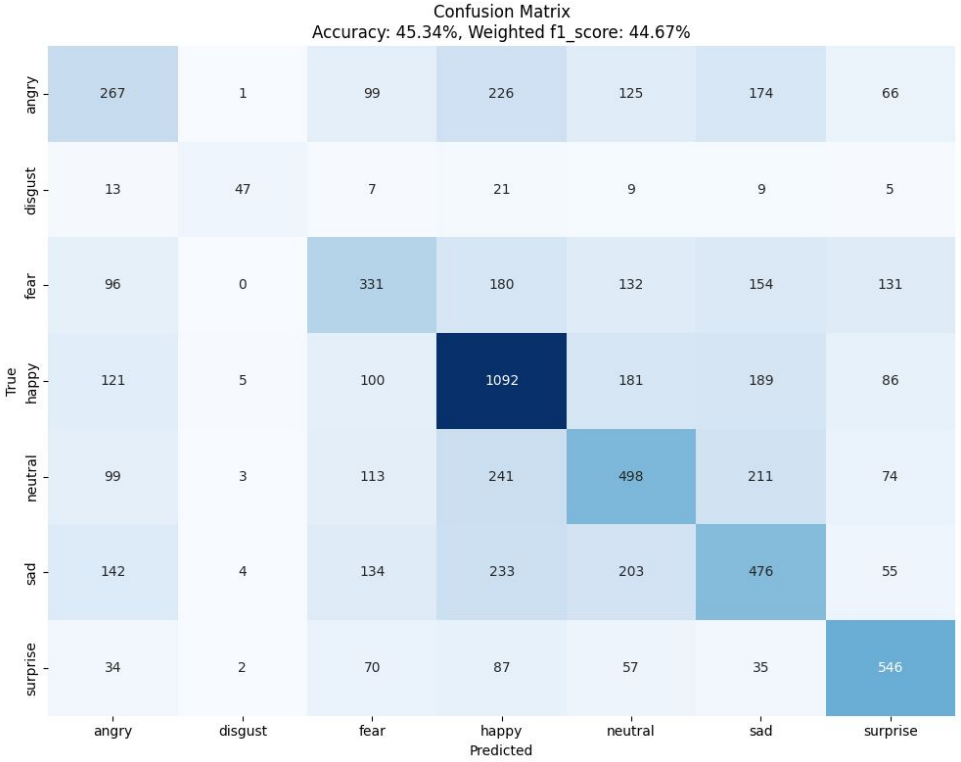


Random Forest Classifier

Final Model

- n estimators = 175
- criteria = gini
- min_samples_split = 3



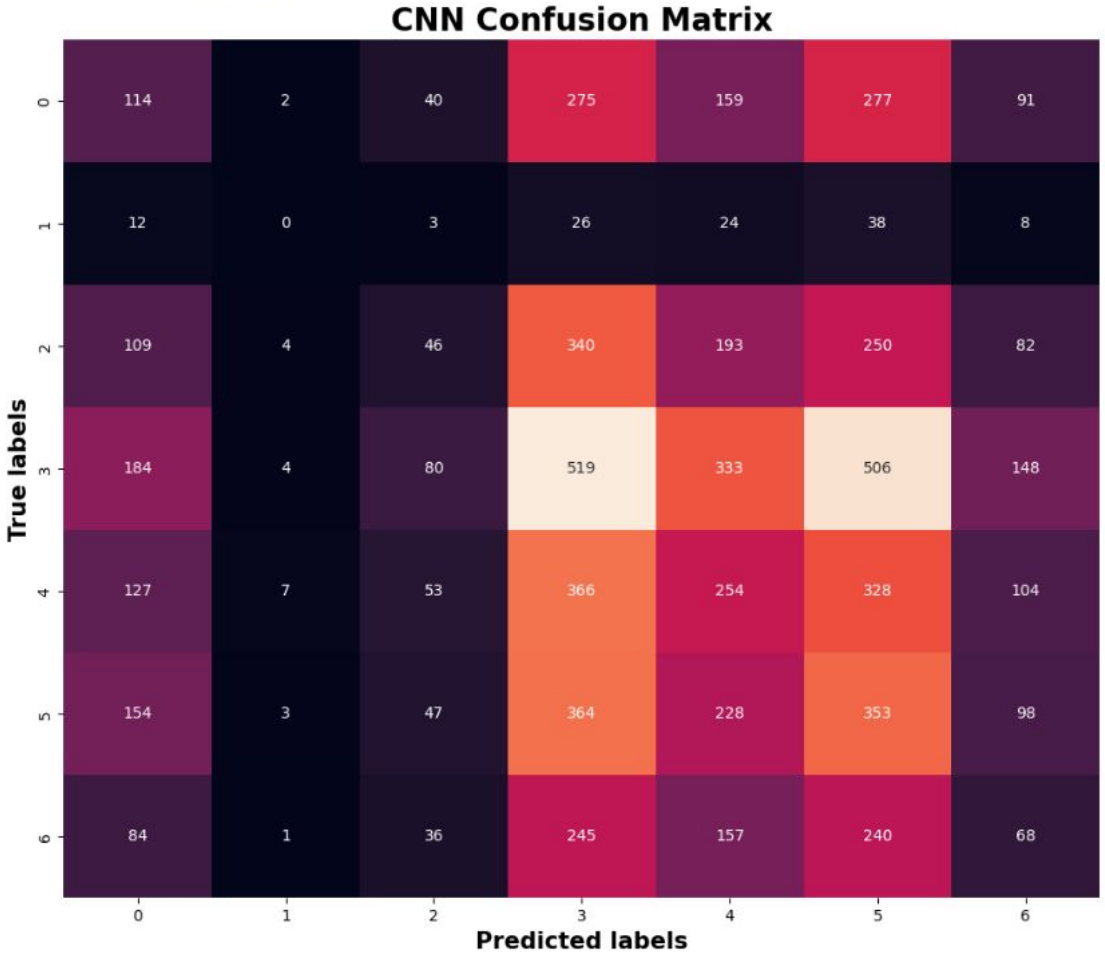


Convolutional Neural Network (CNN)

Model Complexity

	Layer	Parameter	Accuracy
CNN Model 1	4 Conv + 4 Dense	1006791	58.37%
CNN Model 2	2 Conv + 3 Dense	897095	55.21%
CNN Model 3	2 Conv + 3 Dense	243687	52.70%

CNN


CNN Confusion Matrix
True labels
Predicted labels
0 1 2 3 4 5 6
114 2 40 275 159 277 91
12 0 3 26 24 38 8
109 4 46 340 193 250 82
184 4 80 519 333 506 148
127 7 53 366 254 328 104
154 3 47 364 228 353 98
84 1 36 245 157 240 68

Convolutional Neural Network (CNN)

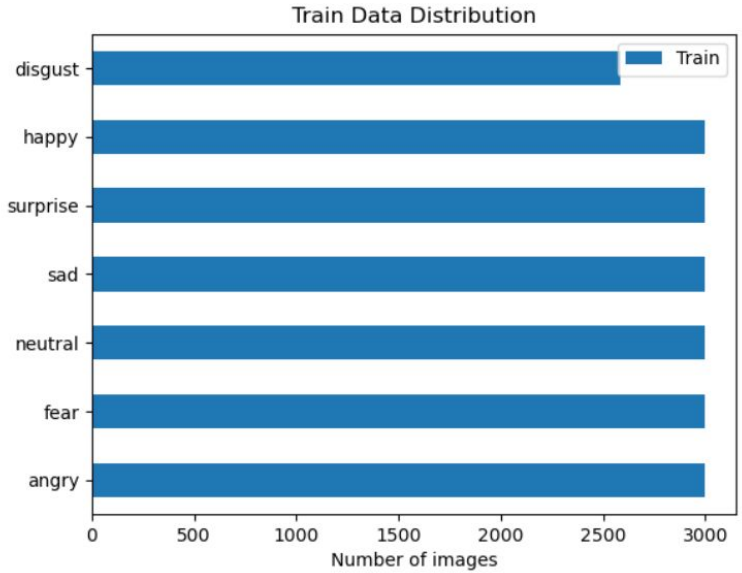
Data Augmentation

- Disgust class increased to 2585 images from 436
- Use 3000 samples from other classes

```
# Define the augmentation parameters
datagen = ImageDataGenerator(
    rotation_range=15,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

Convolutional Neural Network (CNN)

Data Augmentation



Accuracy	
Unbalanced	Balanced
52.70%	50.38%

F1 Score	
Unbalanced	Balanced
14.5%	14.2%