Emotion Recognition

Machine Learning project report

Νικόλας Χελιώτης

Introduction

Emotion recognition is considered one of the hardest and most fascinating topics in computer vision.Emotion expression is complex and subjective.This project focuses on emotion classification using the FER2013 dataset, a widely used dataset containing grayscale images of facial expressions categorized into seven emotion classes: **angry, disgust, fear, happy, neutral, sad, and surprised.**

The objective of this project is to extract meaningful features from facial images and train different machine learning models to classify emotions accurately.The project explores feature extraction techniques, applies various classification models, and conducts hyper-parameter tuning to optimise performance.Only traditional machine learning algorithms such as Support Vector Machines (SVMs) and Random Forest are used and their effectiveness is analysed based on accuracy, precision, recall, and F1-score.

This report is divided into 6 sections:

- **Section 1** provides an overview of how the code was structured.
- **Section 2** delves into the different features extracted from the images, the rationale behind their selection, and the techniques used to optimise them.
- **Section 3** presents the experimentation process using different classifiers.
- **Section 4** focuses on the analysis of the best-performing models, highlighting the optimal hyper-parameters obtained through fine-tuning.
- **Section 5** raises critical questions regarding potential improvements and compares our results with those from previous studies.
- **Section 6** concludes the project's report.

1. Project's technical structure

The project consists of 4 distinct python files, each serving a different purpose:

1. **ExtractData.py**: Loads the dataset and extracts the emotion labels

2. **ExtractFeatues.py** : Processes raw data to extract meaningful features.

3. **Classifiers.py** : Trains machine learning models and fine-tunes them.

4. **GetInsight**: Visualises the results of the classifiers.

Training models is a time consuming task, so each fitted model was saved so it can be easily accessible.

2. Features

Since the dataset only contained images, no features were given so we had to extract our own.The images were all 48x48 pixels and greyscale so no preprocessing was required.

The question was which features to choose.Our first idea was to use Gabor+HOG (Histograms of Oriented Gradients) features.Gabor filters would isolate key facial features such as the eyes, eyebrows, mouth etc.Then we would use HOG on top of the filtered images to extract features.However, this would increase the total number of features by a lot, more precisely 36.000 features per image.Since speed was an important consideration in our project, we dropped Gabor features.

Instead we used a combination of HOG and LBP(Local Binary Patterns).The idea was that HOG features would catch information regarding the face's shape (mouth's angle, eyebrows etc.) and LBP would catch patterns (such as wrinkles around the eyes).

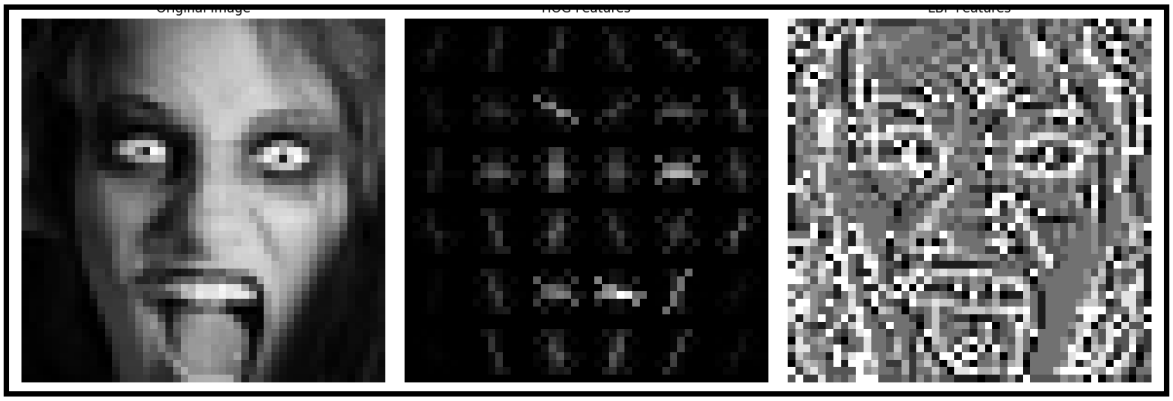

Figure 1. Original Image, HOG features and LBP features.

We ended up with 861 features per image.To optimise even further, we applied PCA keeping 95% of the variance.This resulted in 269 features per image.

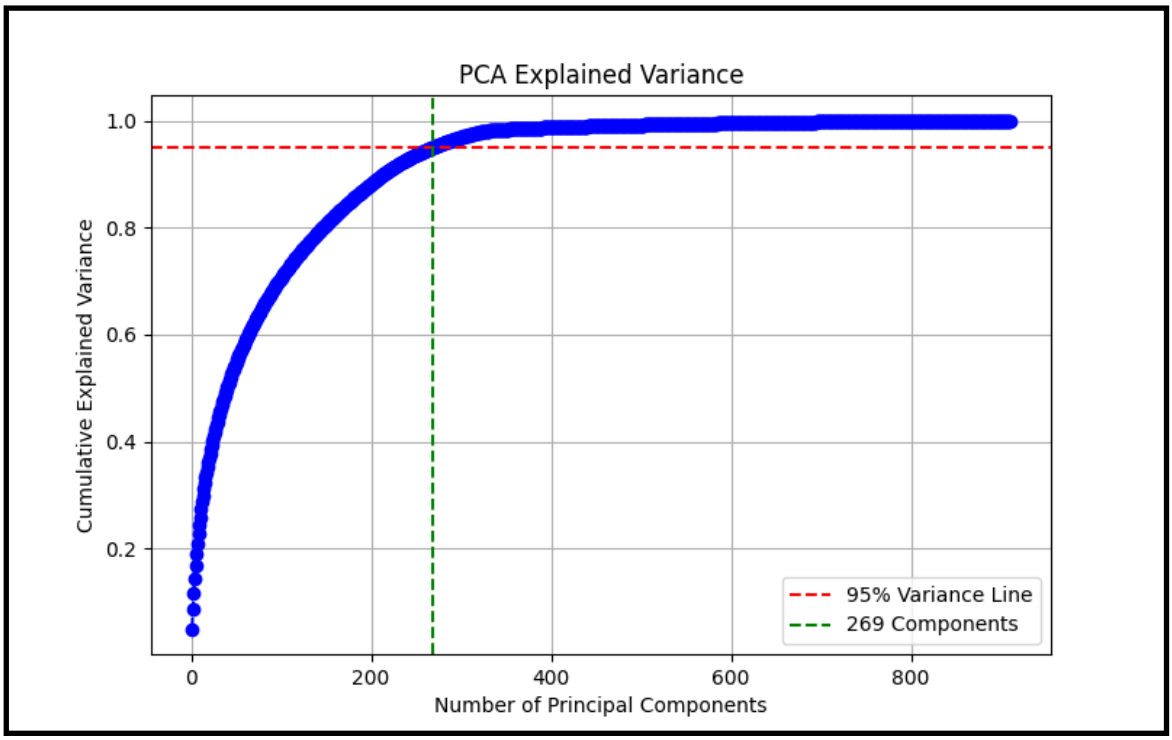

Figure 2. PCA variance curve.

The labels' graph at Figure 3 shows a clear problem.Class 1 is under sampled in comparison with other classes.Class 3, has the most amount of samples.

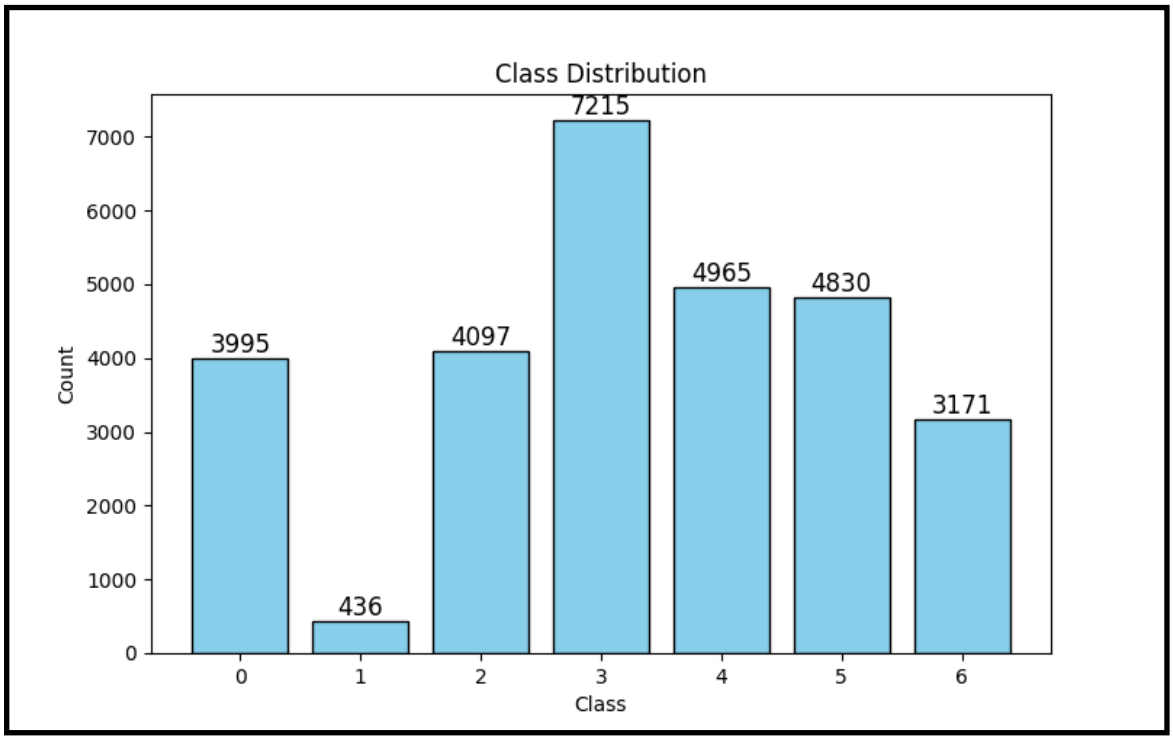

Figure 3. Labels' distribution graph.

3. Experimentation

3.1 Random classifier

Our initial objective was to establish a baseline for performance evaluation. To achieve this, we implemented a classifier that assigns images based on the probability distribution of the existing classes.

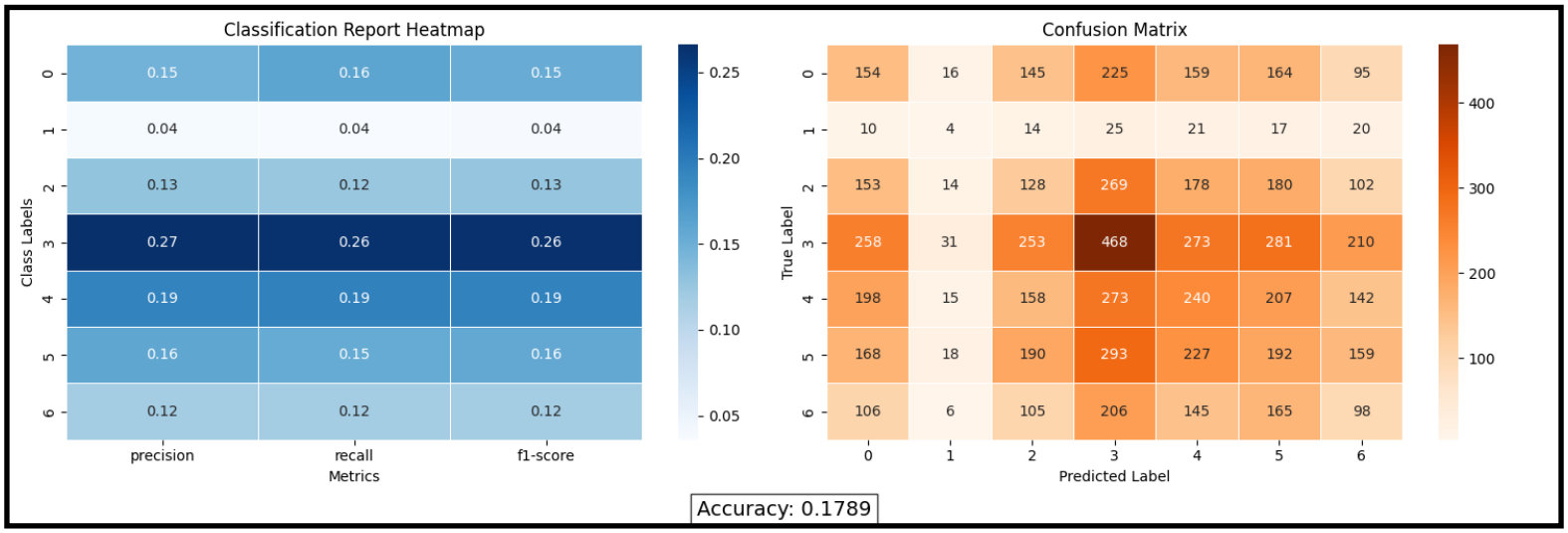

Figure 4. Random classifier's confusion matrix and report.

Macro F1-score	0.15
Macro Recall	0.14
Macro Precision	0.15
Accuracy	0.17%

Figure 5. Random classifier's macro results.

As expected, class 3 (which has the most samples) has the biggest precision and recall score.Class 1 (which has the least amount of samples) has the lowest recall and precision score.

3.2 Svm (linear kernel)

Our next goal was to implement a real classifier and compare its results with those of the random baseline. This comparison helps determine whether our dataset contains meaningful patterns that a model can learn, rather than just noise.For this purpose, we selected a Support Vector Machine (SVM) with a linear kernel.

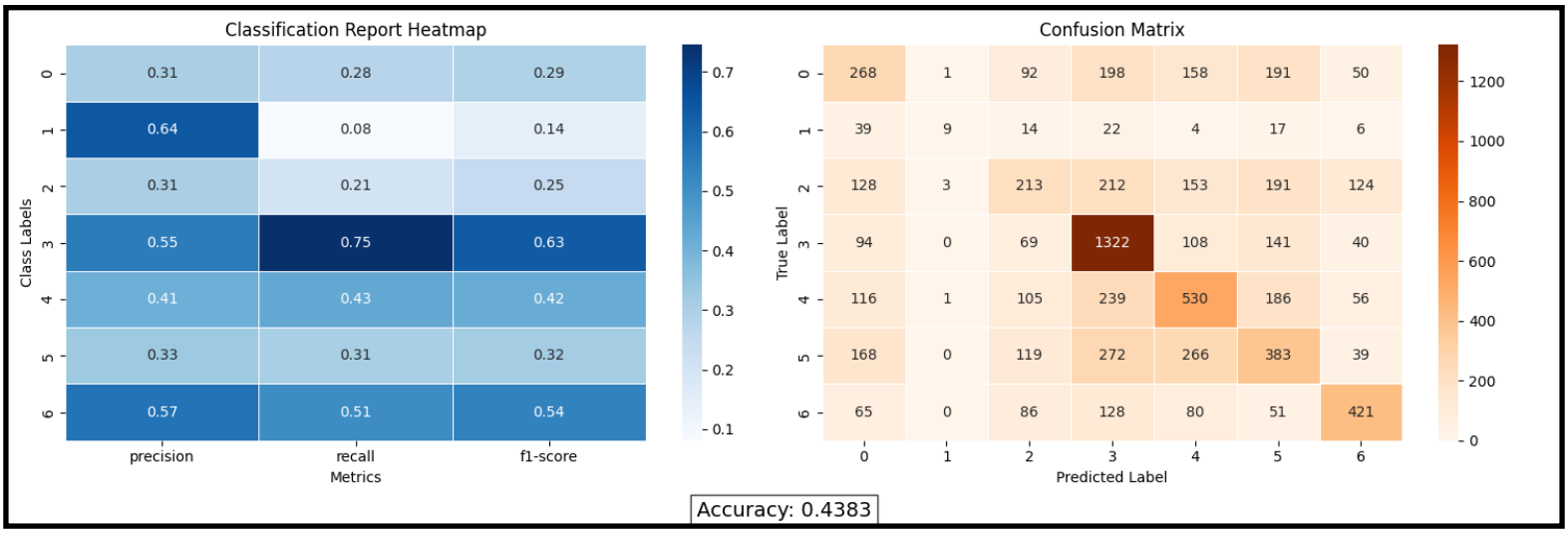

Figure 6. SVM classifier's confusion matrix and report.

Macro F1-score	0.37
Macro Recall	0.36
Macro Precision	0.44
Accuracy	0.43%

Figure 7. SVM classifier's macro results.

The results were highly encouraging. Compared to the random classifier, all evaluation metrics showed significant improvement, confirming that our extracted features contain meaningful information. This also indicates that our SVM with a linear kernel was able to effectively differentiate between classes.However, we observed that Class 1's recall was particularly low (0.08), meaning that the classifier rarely predicts this class correctly.

3.3 Svm (linear kernel, weighted)

The previous experiment showed that class 1 barely gets predicted.To tackle this problem we introduced weights to the classes.The results are shown in Figure 8 and Figure 9.

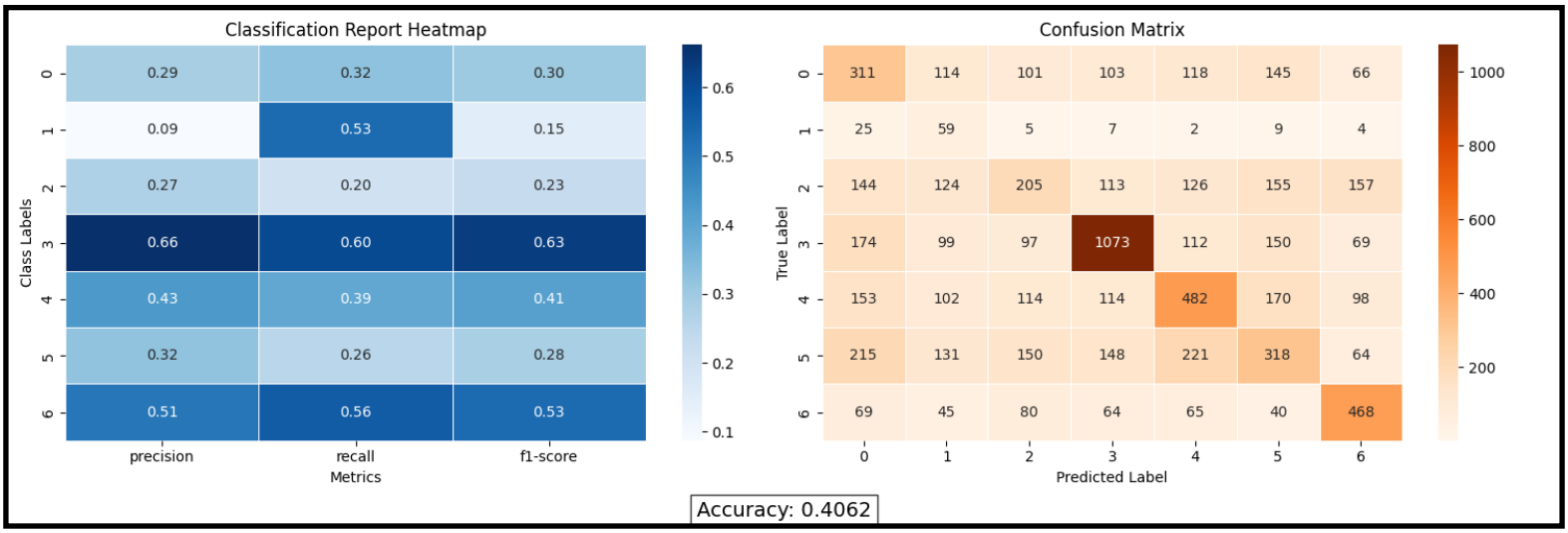

Figure 8. SVM classifier's confusion matrix and report.

Macro F1-score	0.36
Macro Recall	0.40
Macro Precision	0.36
Accuracy	0.40%

Figure 9. SVM classifier's macro results.

Introducing weights increased Class 1 recall.However, Class 1 precision score dropped a lot.This is because SVM now classifies images to Class 1 more often but fails to classify correctly.There is an overall drop in macro scores except macro recall where there is a 4% increase (because Class 1 recall score increased).

3.4 Svm (brf kernel, weighted)

For our next experiment, we used SVM with the RBF kernel to assess whether the data is linearly separable or not. Unlike the linear kernel, which assumes that classes can be separated by a straight hyperplane, the RBF kernel allows for more complex decision boundaries by mapping the data into a higher-dimensional space.

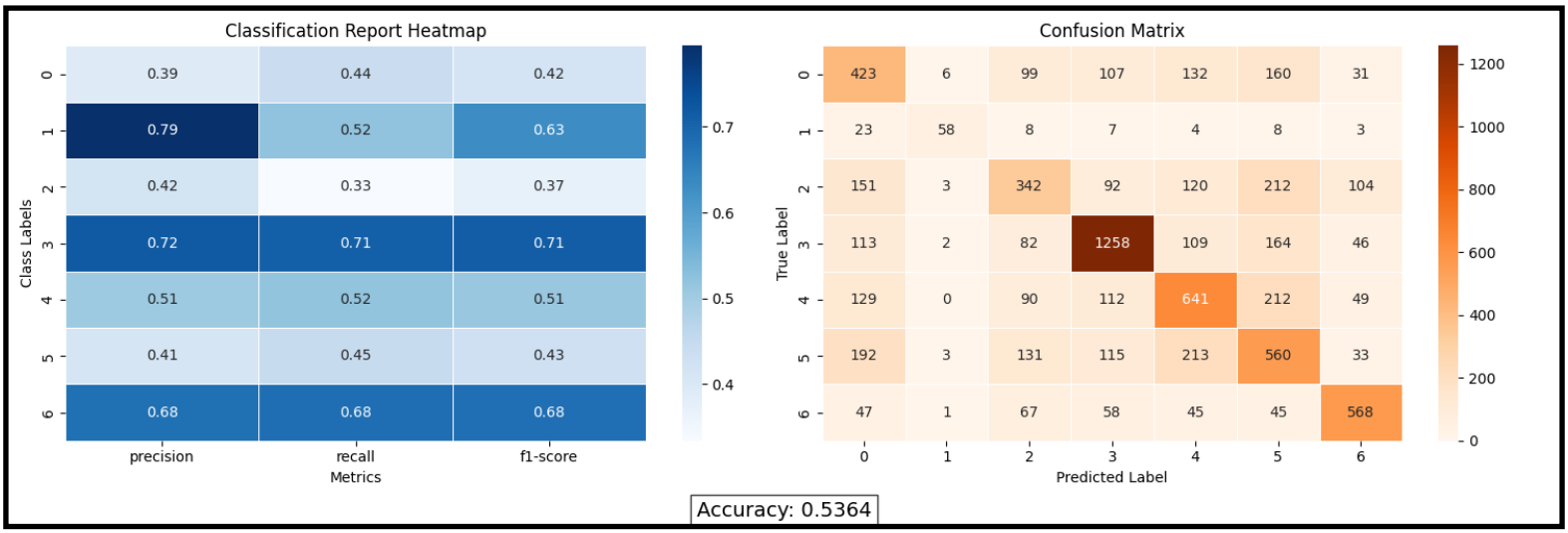

Figure 10. SVM classifier's confusion matrix and report.

Macro F1-score	0.53
Macro Recall	0.52
Macro Precision	0.56
Accuracy	0.53%

Figure 11. SVM classifier's macro results.

The results were highly encouraging. We observed an overall increase in all macro metrics, indicating that the RBF SVM was able to capture more complex patterns in the data compared to the linear SVM.A particularly notable improvement was in Class 1's precision, which not only increased but also achieved the highest precision score among all classes.

3.5 Random Forest

We also experimented with Random Forest.Unlike SVM, which relies on margin-based separation, Random Forest is a decision tree-based method that aggregates multiple decision trees to make more robust predictions.

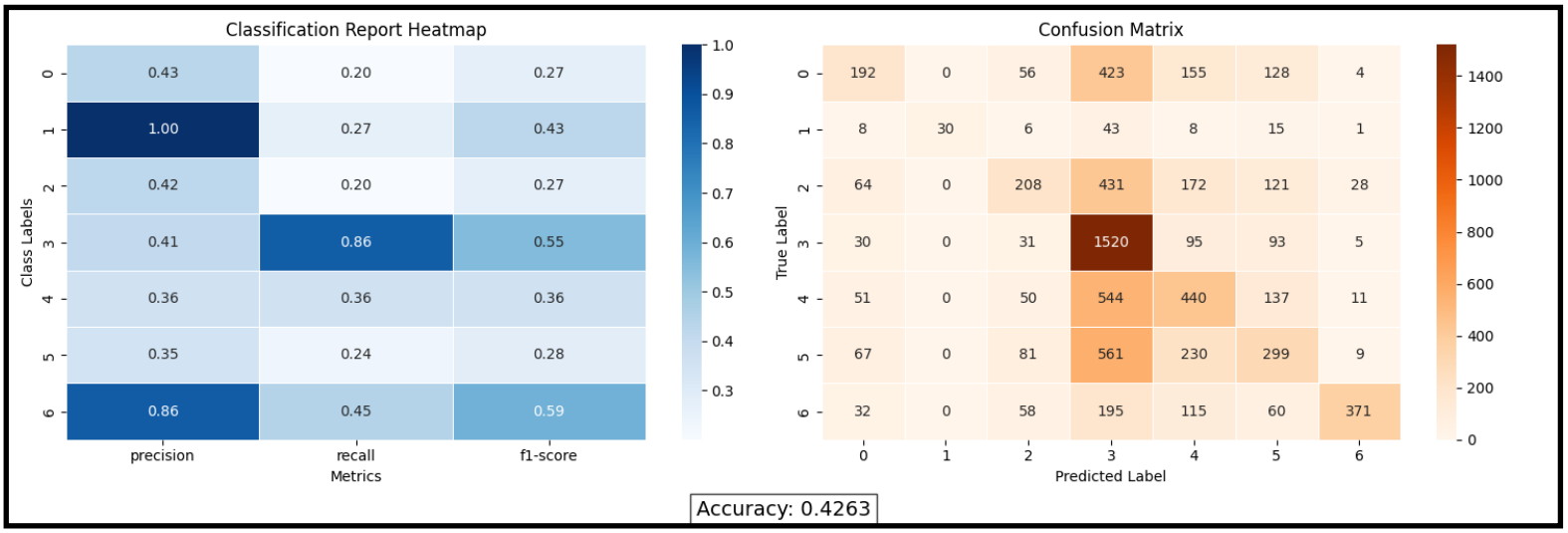

Figure 12. Random Forest's confusion matrix and report.

Macro F1-score	0.43
Macro Recall	0.36
Macro Precision	0.54
Accuracy	0.42%

Figure 13. Random Forest's macro results.

A notable observation from the Random Forest experiment was that, despite Class 1 having a low recall of 0.27, whenever the model classified an instance as Class 1, it was 100% correct (precision = 1.00). This indicates that while the model was highly confident in its predictions for Class 1, it was also extremely selective, rarely assigning instances to this class.

Motivated by this, we experimented with Random Forest using class weights, aiming to improve the recall for underrepresented classes and achieve a more balanced classification. However, the results were disappointing, as none of the key metrics showed any noticeable improvement.

4. Hyper-parameter tuning

Based on the results, we decided to fine-tune the weighted SVM with an RBF kernel and Random Forest to identify the optimal hyper parameters and further improve classification performance.

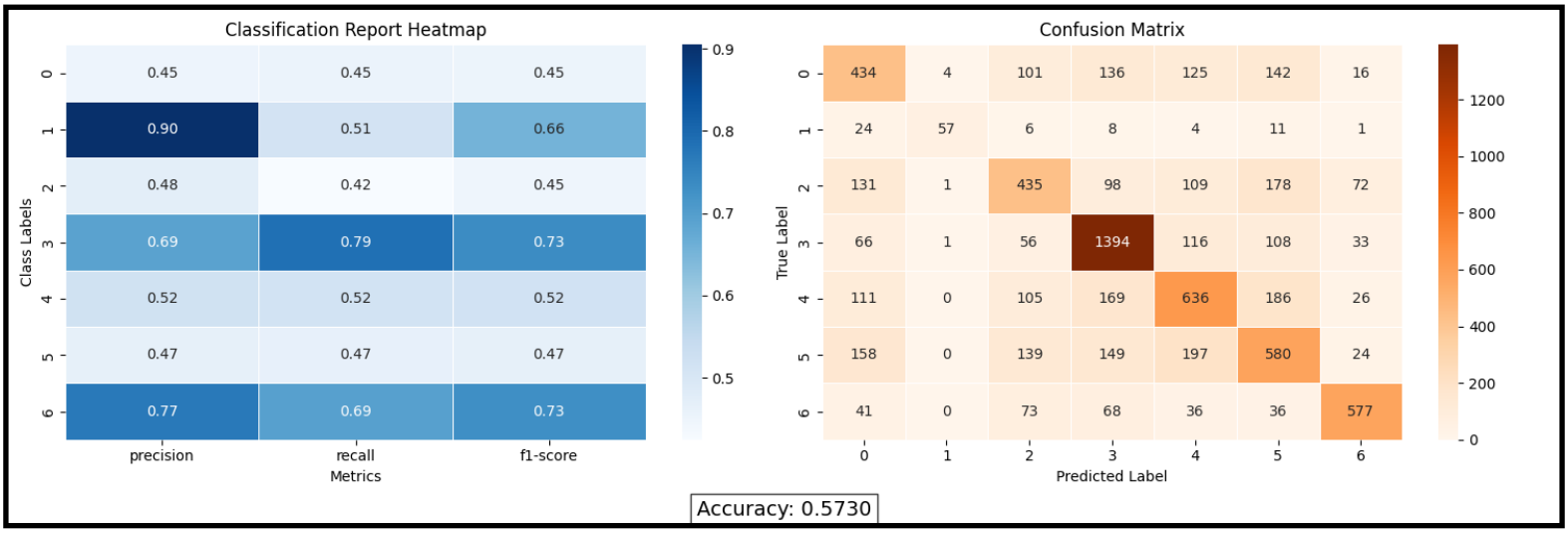

Figure 14. Best SVM's confusion matrix and report.

Macro F1-score	0.57
Macro Recall	0.55
Macro Precision	0.61
Accuracy	0.57%

Figure 15. Best SVM's macro results.

The results were fair, showing an overall improvement in all metrics compared to the previous RBF kernel SVM. Notably, Class 1's precision increased by 9%, indicating that the fine-tuning process helped the model make more confident and accurate predictions for this class.

That was not the case for the best Random Forest model.There was no notable increase in any metric as seen in Figure 17.

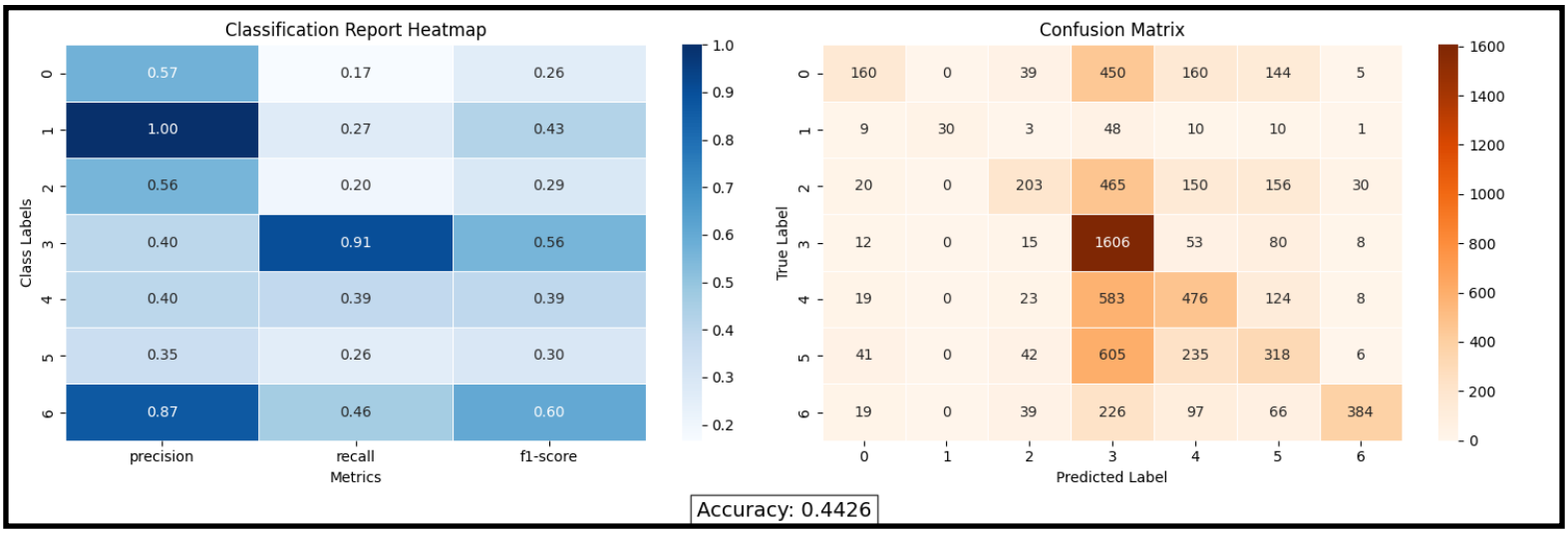

Figure 16. Best RF's confusion matrix and report.

Macro F1-score	0.40
Macro Recall	0.38
Macro Precision	0.59
Accuracy	0.44%

Figure 17. Best RF's macro results.

5. Comparison and Questions

Our results may not seem particularly impressive at first glance, with the best model achieving an accuracy of 57%, F1-score of 0.57, Recall of 0.55 and precision of 0.61.However, comparing with other studies:

1. To our best knowledge, the benchmark (without extra training data) for the FER2013 dataset is **73.28%** (using VGGNet architecture) by Khaireddin.

2. Zhang (2015) reached **75.1% accuracy** by integrating auxiliary data and advanced feature extraction.

3. For the FER2013 dataset, an **accuracy** of **66.3%** is considered very acceptable for CNN methods (Hanya 2024).

We can conclude that, despite relying on traditional machine learning methods, our results are quite acceptable given the complexity of the FER-2013 dataset. While deep learning models such as Convolutional Neural Networks (CNNs) often outperform traditional approaches, our results demonstrate that well-engineered features and careful model tuning can still achieve reasonable performance.To further push these metrics higher, we can explore several key questions:

1.Does age introduce noise? (Older people have wrinkles regardless of whether they are smiling or not)

2.Does ethnicity impact model classification? (Facial structure variations across different ethnicities)

3. Does emotion intensity affect recognition? (Subtle smile vs big smile)

4.Does gender affect accuracy? (Maybe the dataset contains more smiling women, leading the algorithm to learn a biased pattern: *"If* woman→predict smile.")

6. Conclusion

Emotion recognition is one of the most challenging tasks in computer vision, as facial expressions can be highly variable due to differences in lighting, angles, occlusions, and individual facial structures. Despite these challenges, our study demonstrated that traditional machine learning methods can still achieve reasonable performance when combined with effective feature extraction and model tuning.

Through our experiments, we observed that while classifiers like SVM with an RBF kernel and Random Forest provided meaningful improvements over a random baseline, their overall accuracy remained limited, with our best model achieving 57% accuracy, an F1-score of 0.57, recall of 0.55, and precision of 0.61. These results align with the expected performance of traditional ML models on FER-2013, highlighting the inherent difficulty of the task.

We hope to revisit this project in the second semester with a fresh perspective and new techniques to further push the metrics higher. By leveraging more advanced methods, such as deep learning architectures and improved feature extraction techniques we aim to enhance our classification performance beyond the current results.