Detecting Fake Images

Team Members: Heba Bou KaedBey, Navid Bahadoran, Rajeev Gopeesingh, Chase Wiederstein, and Jonathan Engle

GitHub: `https://github.com/hebabkb/FakevsReal`

Problem Description: We are addressing fake image detection in news media. With the rise of misinformation, there has been an increasing challenge in identifying manipulated or fake images used in news outlets and social media platforms to mislead the public. These fake images can contribute to the spread of false information, and cause societal harm.

We tackle this problem by developing a machine learning model to accurately detect fake images from a dataset using various image features. This model is trained on labeled data (real vs. fake) to learn the subtle differences between real and tampered images, using features like glcm, ela, wavelet transforms, color histogram and edge.

Stakeholders for the Project: News Agencies, Social Media Platforms, Government and Law Enforcement Agencies, Advertising companies, Journalists and Reporters, Fact-Checking Organizations, Social Media Users.

Data Collection: Our model is trained on a publicly available dataset of 12614 images. This dataset contains 7491 real images and 5123 fake images.

Data Preparation: To enhance the model's ability to distinguish between real and fake images, several feature extraction techniques were employed: ELA, Wavelet Transforms, GLCM, Color Histogram, and Edge Detection.

Data Balancing, Scaling and Dimensionality Reduction:

- **Balance**: Given the initial distribution of images, SMOTE was applied to balance the dataset, ensuring equal representation of both classes.
- **Scale**: Robust Scaler was used to standardize the feature set.
- **Dimensionality Reduction**: Used LLE.

Model Approach: Out of the 11 models that we tested we chose the models with the highest PR AUC and ROC AUC. The reason why PR AUC is important is because PR AUC emphasizes precision (how many detected fakes are actually fake) and recall (how many true fakes are detected). These metrics are directly tied to the goal of minimizing false positives and false negatives in detecting fake images. The reason why ROC AUC is important since it balances TPR and false positive rate (FPR), providing a measure of the overall separability between real and fake images.

Model Architecture Overview: The first chosen model is a single-level stacking classifier featuring SVM as meta-learner.

- Level 1 (Base Models):Four classifiers: Random Forest, XGBoost, LightGBM, and SVM are used as the base models. These classifiers offer complementary strengths: Random Forest provides robust ensemble learning with limited overfitting, XGBoost specializes in handling complex decision boundaries with gradient boosting, LightGBM optimizes speed and accuracy

for large datasets, and SVM ensures reliable performance in high-dimensional feature spaces.

- Level 2 (Meta-Learner): The outputs of the base models are aggregated by a meta-learner SVM. This final estimator synthesizes predictions to deliver a cohesive and accurate classification, leveraging SVM's ability to discern complex patterns and relationships.

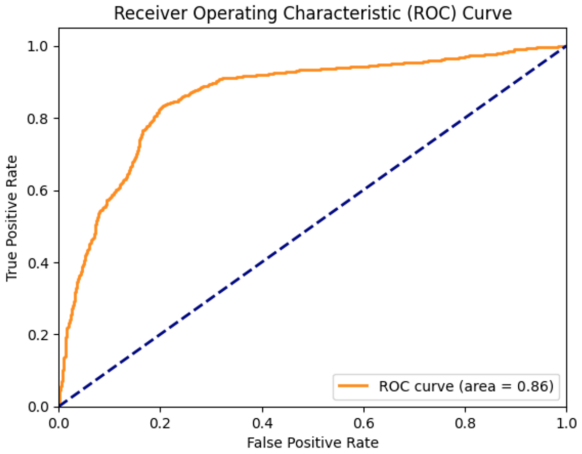
Feature Engineering: The input data is preprocessed with LLE for dimensionality reduction, which enhances model efficiency and accuracy. The training dataset is balanced using SMOTE to address class imbalance, ensuring fair representation of both real and fake images.

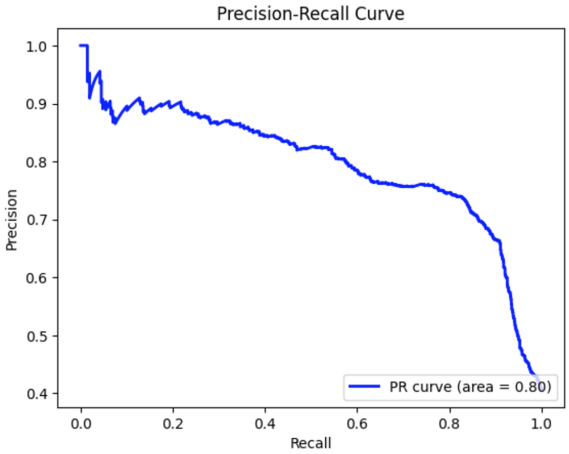
Model Evaluation: The model is trained on a balanced dataset and evaluated using metrics such as accuracy, precision, and recall emphasizing the detection of fake images. The stacking classifier achieves an overall accuracy of 81%. Future improvements aim to enhance precision further and improve the generalization to unforeseen fake image samples.

```
SVM as Meta-Learner Accuracy: 0.8097502972651606
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.80      0.83      1498
           1       0.74      0.83      0.78      1025

    accuracy                           0.81      2523
   macro avg       0.80      0.81      0.81      2523
weighted avg       0.82      0.81      0.81      2523
```





Model Architecture Overview: Second chosen model employs a two-level stacking architecture to improve fake image detection performance.

- Level 1 (Base Models): Four diverse classifiers: Random Forest, XGBoost, LightGBM, and Support Vector Machine (SVM) serve as the foundation. Each base model brings unique strengths: Random Forest offers robust ensemble predictions, XGBoost and LightGBM excel in gradient-boosted tasks, and SVM performs well in high-dimensional spaces.

- Level 1 (Meta-Learners): Predictions from the base models are further refined using three separate meta-learners: LightGBM, SVM, and XGBoost. These meta-learners learn to weigh and combine the outputs of the base models, enhancing predictive accuracy.

- Level 2 (Final Stacking): The outputs from the Level 1 meta-learners are then combined in a second stacking operation. The final meta-learner SVM integrates these predictions into a single cohesive output, leveraging SVM's reliability in classification tasks.

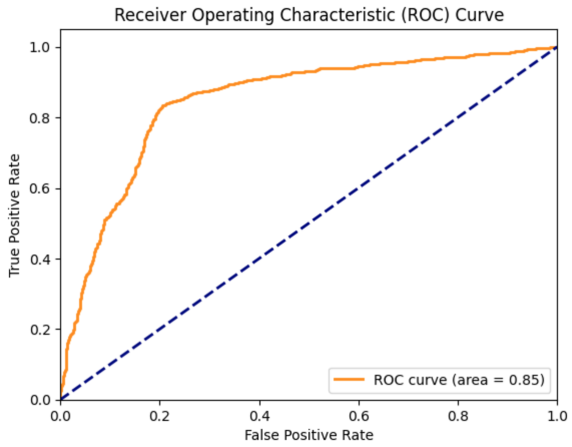
Feature Engineering: The input data is preprocessed with LLE for dimensionality reduction, which enhances model efficiency and accuracy. The training dataset is balanced using SMOTE to address class imbalance, ensuring fair representation of both real and fake images.

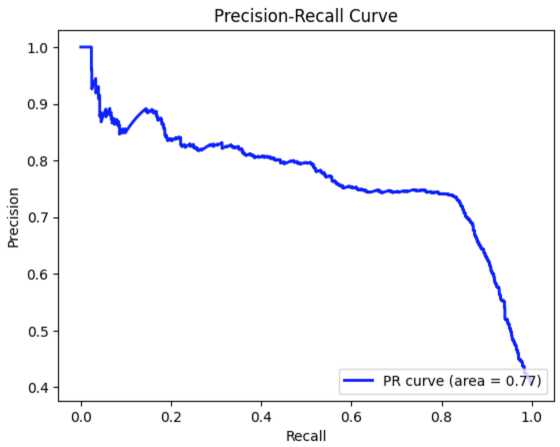
Model Evaluation: The model is trained on the balanced dataset and evaluated using metrics such as accuracy, recall and precision, particularly emphasizing the detection of fake images. Despite achieving an overall high accuracy (81%), further work focuses on improving the precision for fake image detection and ensuring generalization to unseen data.

```
LightGBM as Meta-Learner Accuracy: 0.8097502972651606
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.80      0.83      1498
           1       0.74      0.83      0.78      1025

    accuracy                           0.81      2523
   macro avg       0.80      0.81      0.81      2523
weighted avg       0.82      0.81      0.81      2523
```





Finally, we have written a web app that showcases our model.

Future Iterations: We plan to imporve our model by potentially introducing engineered features that represent combinations or interactions of existing ones. We also want to explore new features that might capture nuances specific to image tampering like some other texture-based features (Gabor filter responses, Entropy, etc...).

Also, we plan to augment the dataset with more diverse examples of tampered images to improve generalizability.

Moreover, penalize the model for errors and reward it for correct classification.
Furthermore, we plan to combine traditional features with deep learning models (CNN- based embeddings for example).