AI Image Detection

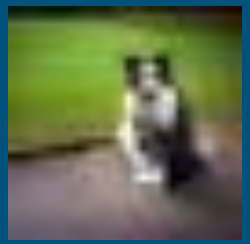
By Terin Ambat, Gabriel Clarence, Soham Joshi

Problem

- Synthetic images are becoming highly realistic
- Humans detect fakes only ~53% accurate
- Misuse risk: misinformation, fraud, identity theft
- Rapid growth of AI-generated content online
- Need automated detection systems

Dataset - CIFAKE

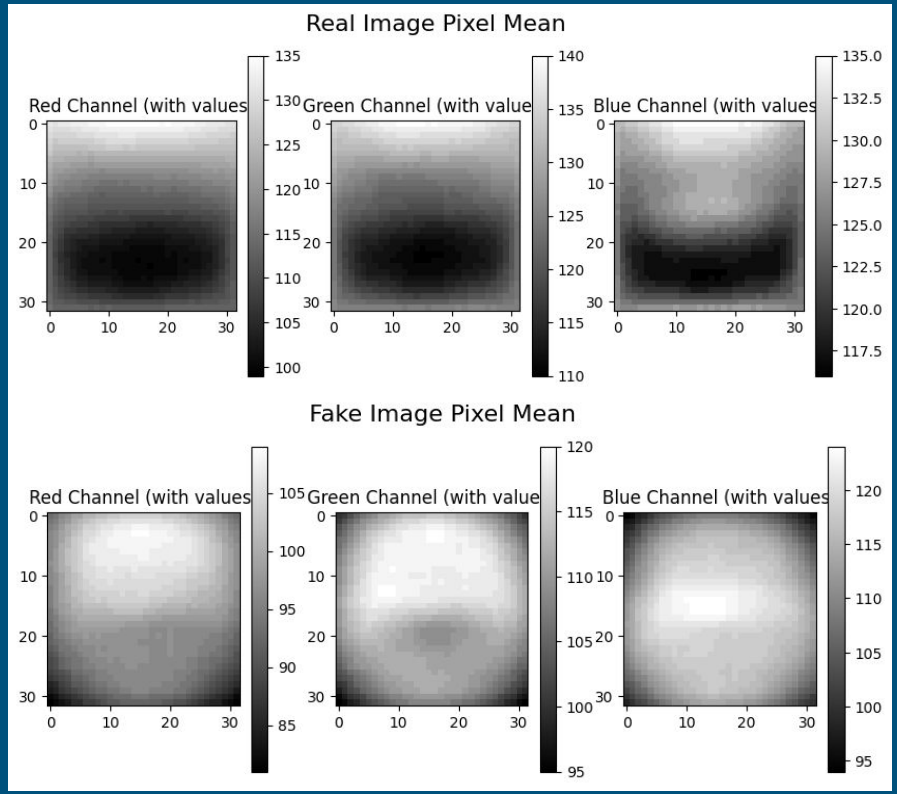
- Based on CIFAR-10
  - Used for training image classification
- Stable Diffusion v1.4 to generate fake image
- 60k Real, 60k Fake
  - 32x32 RGB
  - 10 Categories of image

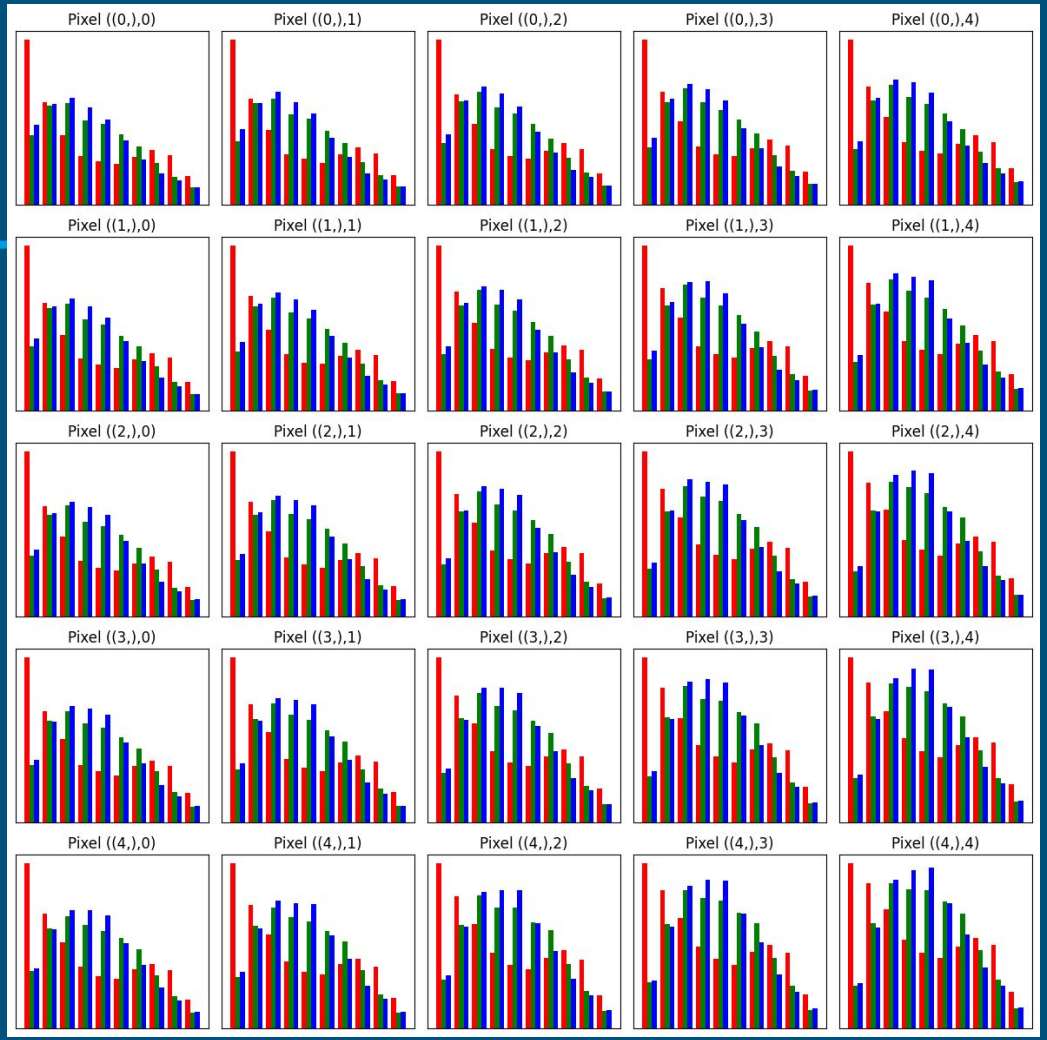


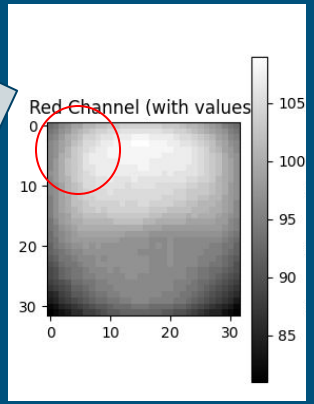
Dataset - EDA

The mean value for each pixel

AI generated Image has a vignette



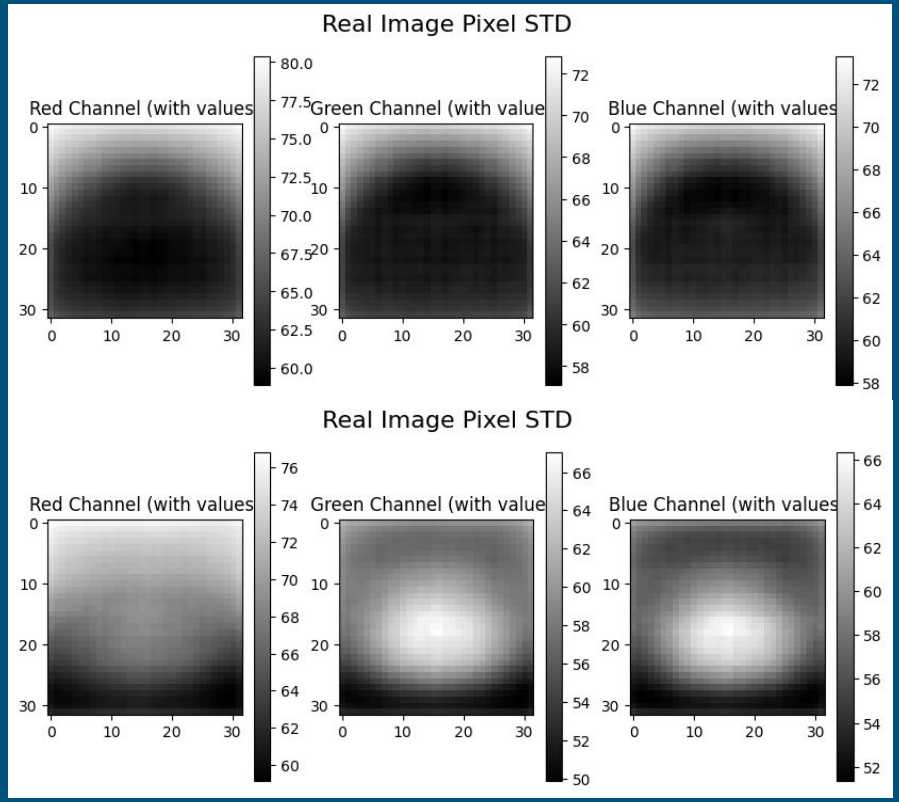

Pixel ((0,),0)
Pixel ((0,),1)
Pixel ((0,),2)
Pixel ((0,),3)
Pixel ((0,),4)
Pixel ((1,),0)
Pixel ((1,),1)
Pixel ((1,),2)
Pixel ((1,),3)
Pixel ((1,),4)
Pixel ((2,),0)
Pixel ((2,),1)
Pixel ((2,),2)
Pixel ((2,),3)
Pixel ((2,),4)
Pixel ((3,),0)
Pixel ((3,),1)
Pixel ((3,),2)
Pixel ((3,),3)
Pixel ((3,),4)
Pixel ((4,),0)
Pixel ((4,),1)
Pixel ((4,),2)
Pixel ((4,),3)
Pixel ((4,),4)


Red Channel (with values
0
10
20
30
0
10
20
30
105
100
95
90
85

Dataset - EDA

High Standard Deviation

Tried to use z value and KNN
- Did not work



Feature Engineering

Error Level Analysis (ELA)

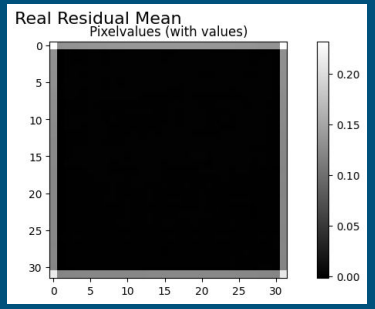
- Compare original with recompressed image (abs error)

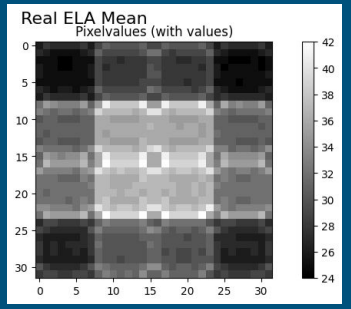
Residual

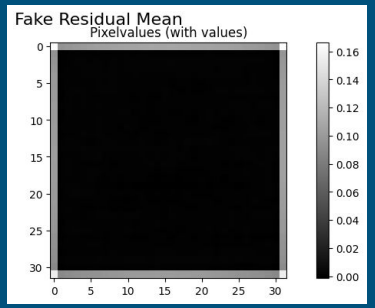
- Image - denoised(image)

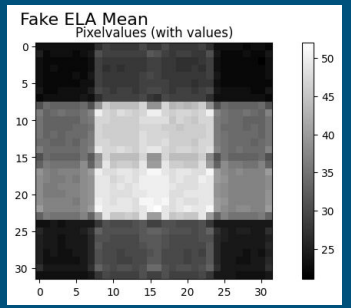
Both need high-res image

- Does not work in our case!









Solution - Overview

- General Pre-Processing:
  - Recombining Train/Test Datasets & Shuffling
  - Normalizing data to [0, 1]
  - Standardizing Data
  - Flattening Images (32 x 32 x 3) → (3072)
- Models
  - CNN
  - KNN
  - Logistic Regression
  - SVM
  - XGBoost

- Main Metric: **F1 Score**
  - Predict fake but real(false pos)
    - People get tricked
  - Predict real but fake (false neg)
    - People credibility is damaged

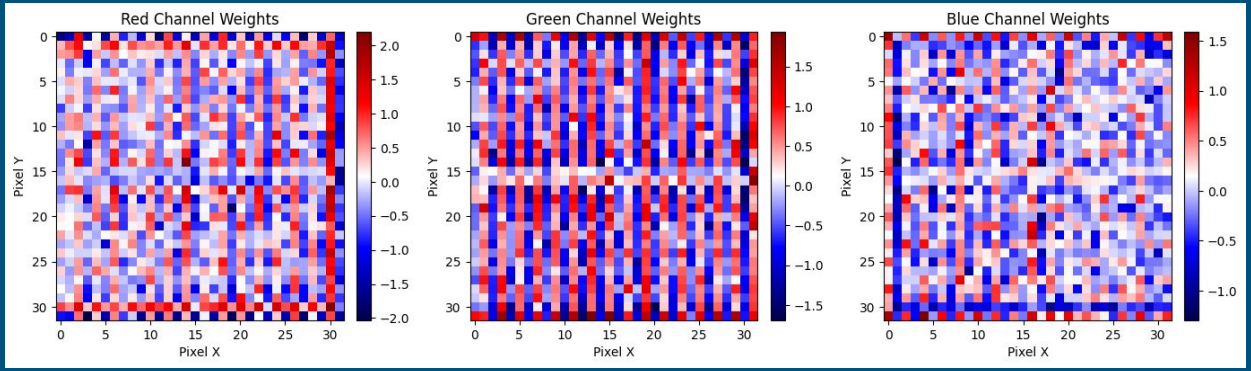
Solution - CNN

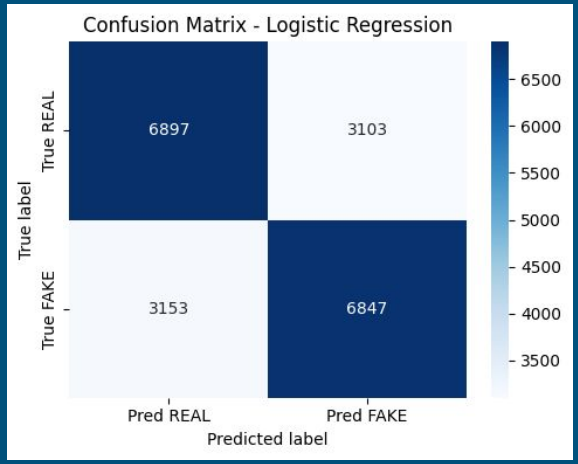
- Motivation: CNN Captures Non-linear Signal Features from Image
- Architecture - Taken from CIFAKE Paper
- Target Model
- Results
  - F1: **93.5%**

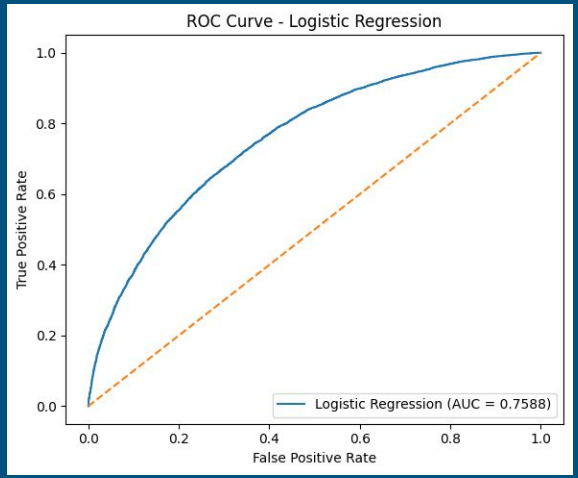
Layer (type)	Output Shape	Param #
rescaling (Rescaling)	(None, 32, 32, 3)	0
conv2d (Conv2D)	(None, 30, 30, 32)	896
max_pooling2d (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_1 (Conv2D)	(None, 13, 13, 32)	9,248
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 32)	0
flatten (Flatten)	(None, 1152)	0
dense (Dense)	(None, 64)	73,792
dense_1 (Dense)	(None, 1)	65

Solution - Logistic Regression

- Uses flattened pixel values (3072 features)
- Simple linear classifier
- Fast to train, easy to interpret
- Serves as baseline for comparison
- Achieved ~69% Test F1
  - 70.14% Train F1





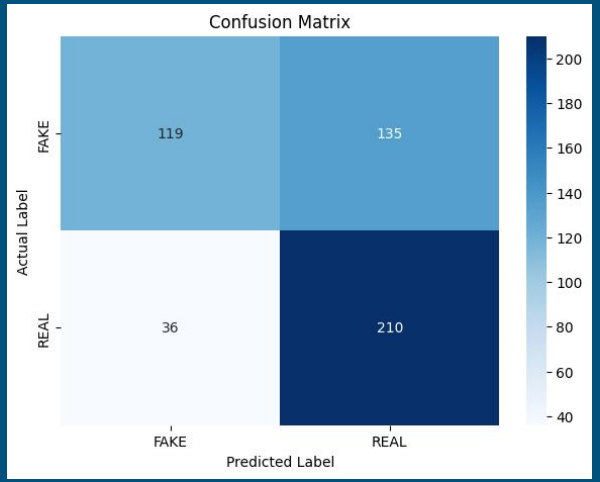


Solution - KNN

F1: 58.19%

Precision: 76.8%

Recall: 46.85%



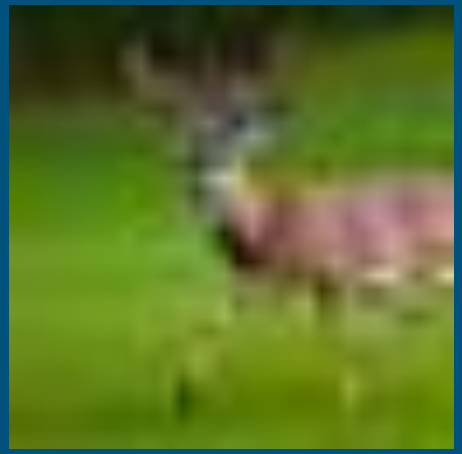
Training Data Stored: 100,000

K: 100

Distance: L2 Euclidean

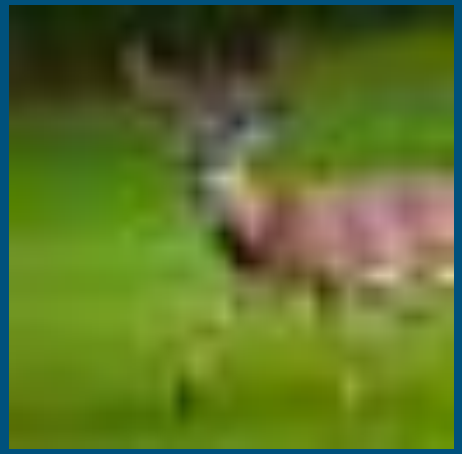
L2: $||x-y||_2$

Why KNN does not perform as good as CNN?



Real

Why KNN does not perform as good as CNN?



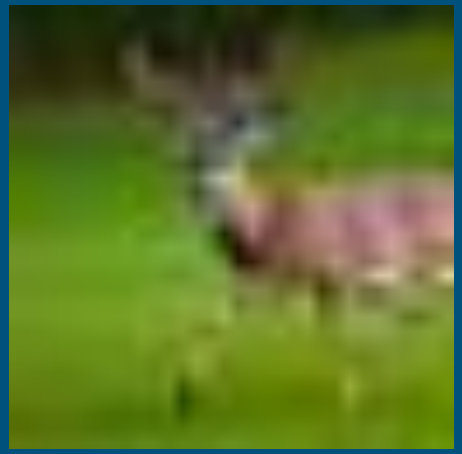
22.9 e6

Real

26.5 e6

Why KNN does not perform as good as CNN?

Fake



22.9 e6

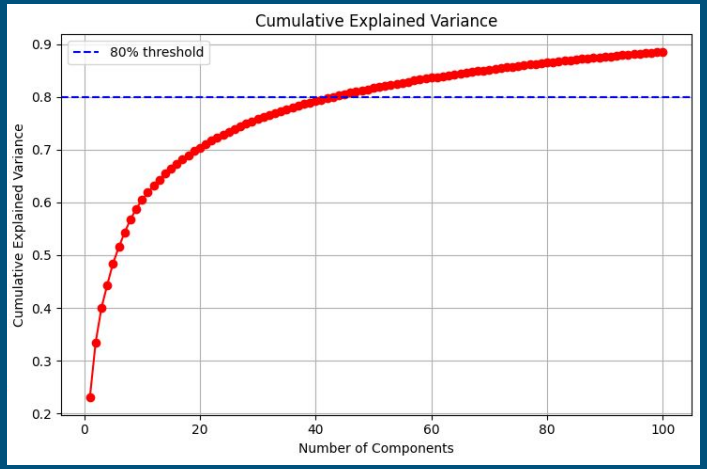
Real

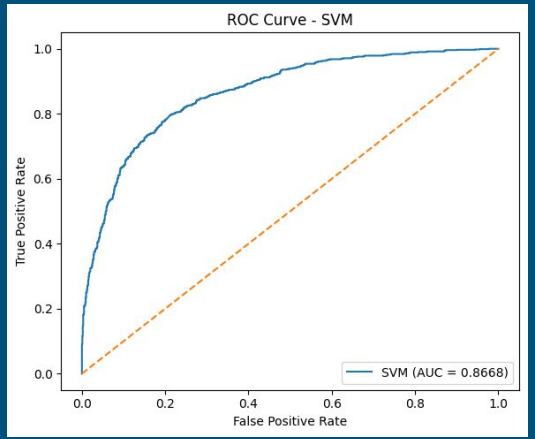
Real

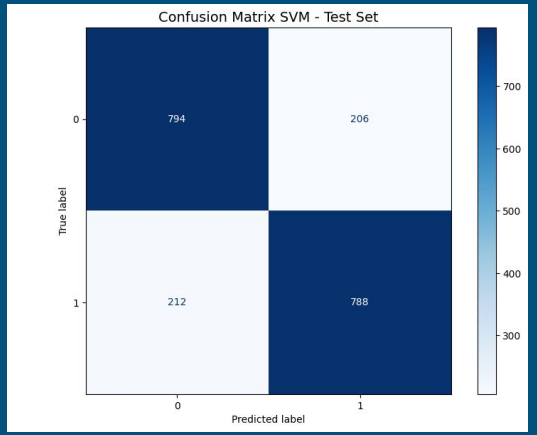
26.5 e6

Solution - SVM

- Motivation: Can capture non-linear patterns with decision boundaries
- Notable Techniques:
  - PCA to Compress Dimensions and Reduce Computational Complexity
  - 3 Fold CV
- Train F1: 84.48%
- Test F1: 79.10%

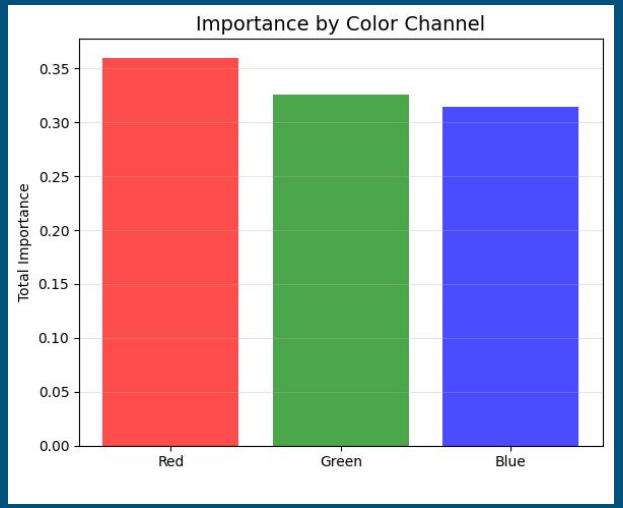


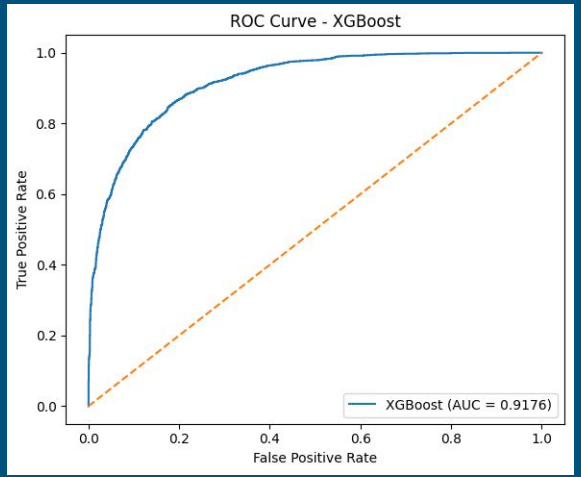


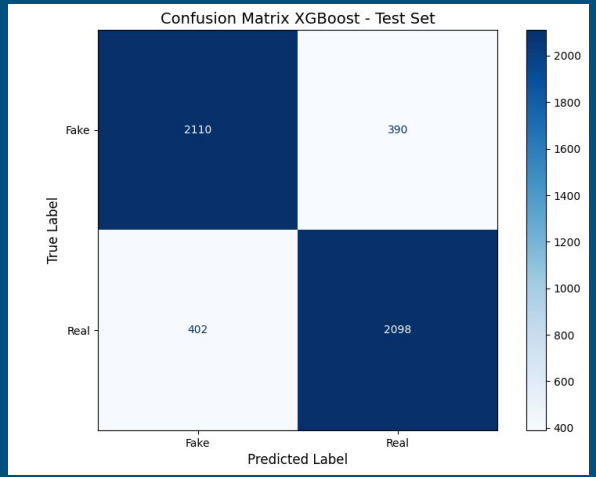


Solution - XGBoost

- Motivation: Can capture non-linear patterns without manual feature engineering
- Notable Techniques
  - High L2 Regularization
  - Validation Monitoring
- Train F1: 95.68%
- Test F1: 84.16%







Comparisons

	Train F1 Score	Test F1 Score
KNN	-	58.19%
Logistic Reg	70.14%	68.72%
SVM	84.48%	79.10%
XGBoost	95.68%	84.16%
CNN	95.70%	93.50%

Limitations / Next Steps

- Models are Trained on Small Images (32x32x3)
  - Not easily Generalizable to All AI Models
- AI Images in dataset are outdated
  - It is only generated with 1 type of model
  - Not diverse
- Only 10 category of images
- Use a larger data set with from variety of data AI generators
- Use CNN and deep learning models

Demo

Github

https://github.com/gclarence011/257-GroupProject-Repo/

References

- Mohammed et al., 2024 (Human Performance in Detecting Deepfakes)
- Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.
- Bird, J.J. and Lotfi, A., 2024. CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images. IEEE Access.