Image Classification: AI-Generated and Real Images

Group Members:

- Sanya Madan (2021561)
- Parisha Agrawal (2021270)
- Brinda Muralie (2021140)

INDRAPRASTHA INSTITUTE *of* INFORMATION TECHNOLOGY **DELHI**

Motivation

- Differentiating between real and AI-generated images has become a complex task for humans due to the remarkable progress in AI image generation.

- Misleading images that convincingly imitate reality can cause significant harm like spreading fake news, damaging reputations, and manipulating public opinion through media.

Literature Review
CIFAKE: Advancing AI-Generated Image Recognition

- The paper addresses the challenge of distinguishing between real-life photographs and AI-generated images.
- A synthetic dataset mirroring the CIFAR-10 dataset is generated, providing a contrasting set of images for comparison.
- The study proposes using a Convolutional Neural Network (CNN) for binary classification of the images into ‘Real’ or 'Fake’.
- After training 36 network topologies, the optimal approach achieved a classification accuracy of 92.98%.
- The study implements explainable AI to identify features useful for classification, focusing on small visual imperfections in the image backgrounds.

Literature Review: GenImage
Advancing AI-Generated Fake Image Detection

- The paper introduces the GenImage dataset, a million-scale benchmark for detecting AI-generated images.
- The dataset includes over one million pairs of AI-generated fake images and real images.
- It covers a broad range of image classes and uses state-of-the-art generators, including advanced diffusion models and GANs.
- The paper proposes two tasks for evaluating detection methods: cross-generator image classification and degraded image classification.
- The GenImage dataset allows researchers to expedite the development and evaluation of superior AI-generated image detectors.

Dataset Description

- Dataset includes 120,000 images, evenly split between real and synthetic (fake) images, categorized into ten distinct classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck, with 6,000 images per class)

- The dataset is divided into 100,000 training images (50,000 each for real and fake images) and 20,000 testing images (10,000 each for real and fake images), all in RGB format and resized to 32x32 pixels.

- The real images used in the dataset were extracted from the CIFAR-10 dataset introduced by Krizhevsky and Hinton. The synthetic images, on the other hand, were generated using Hugging Face's Stable Diffusion Model Version 1.4.

Dataset Visualisation

- TSNE plot (Figure 1) unsuitable due to scattered real and fake images.
- Pixel intensity histogram (Figure 2) does not display much differences between real and fake images.

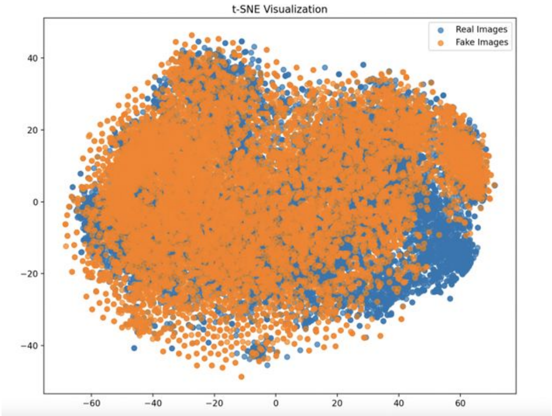


Figure 1. t-SNE Visualisation

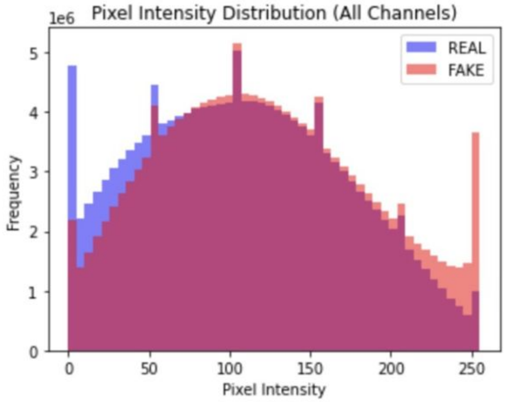


Figure 2. Pixel Intensity Histogram

Dataset Visualisation

- The background defects (Figure 4) highlights variations in background quality, with real images having fewer defects and fake images showing a higher frequency, suggesting potential generation artifacts. These features aid in distinguishing real and fake images based on blur and background characteristics.

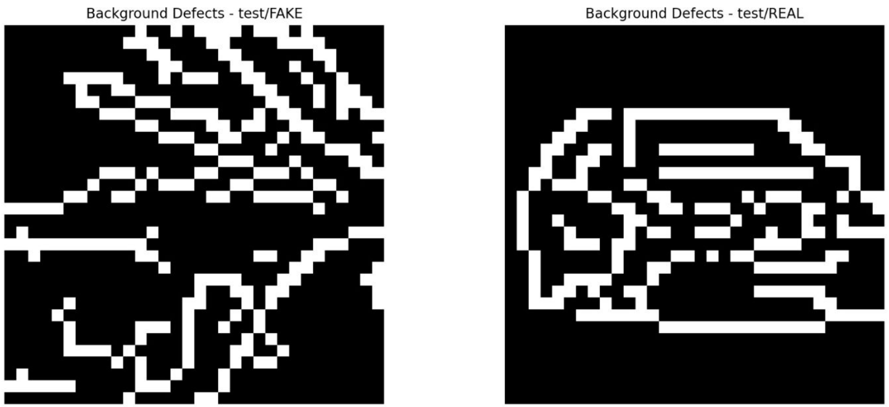


Figure 4. Background Defects

Dataset Visualisation

- The blur level histogram (Figure 3) reveals that real images generally have lower blur levels, preserving finer details, while fake images concentrate at higher blur levels, suggesting potential differences in quality or generation methods.

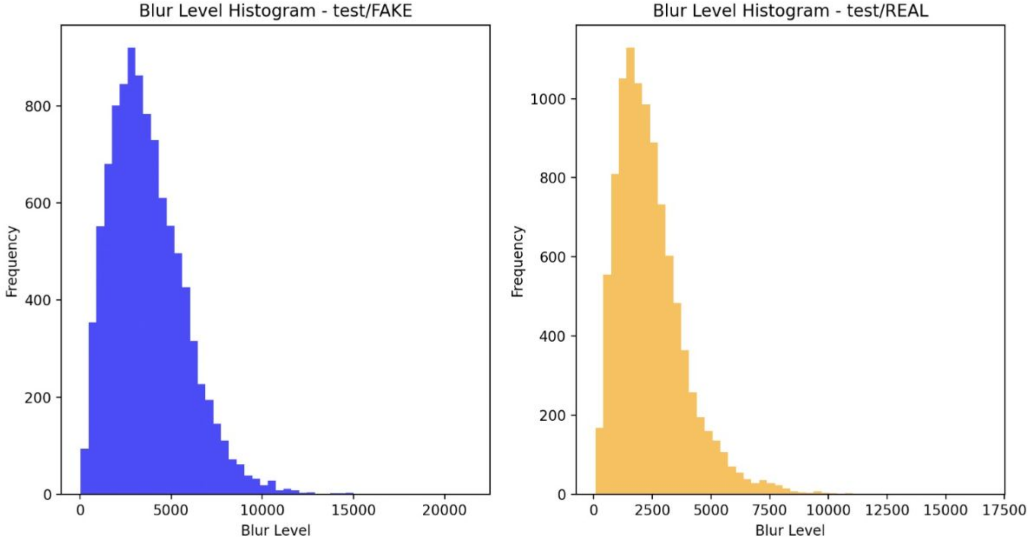


Figure 3. Blur Level Histogram

Dataset Preprocessing

- Images loaded using CV2's 'imread' function and converted to numeric data with Numpy.
- Uniform 32x32 pixel dimensions ensured using OpenCV.
- Class labels transformed to numeric values using scikit-learn's LabelEncoder.
- No outliers found via data visualization and box plot (Figure 6).
- PCA deemed unnecessary as the dataset already had optimal dimensionality which was verified by experiments.

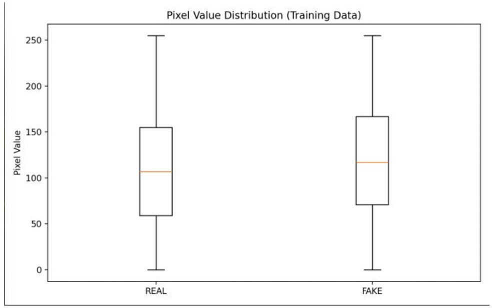


Figure 6. Box Plot

Number of components	Accuracy
10	0.71375
50	0.795
100	0.81175
500	0.81125
1000	0.81325
1024 (32 × 32)	0.8135

Table 1. PCA components and accuracy

Methodology

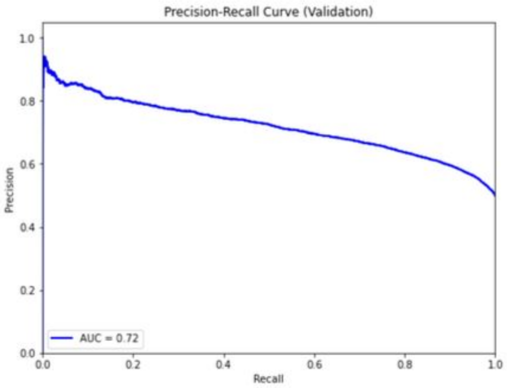
- Employed Logistic Regression, Naïve Bayes classifier, Decision tree Classifier, Random Forest Classifier, Support Vector Machine, Multilayer Perceptron and Convolution Neural Network.
- We have used 7 models to train the dataset and evaluated accuracy scores for each model to compare them.
- Used Scikit learn, matplotlib, TensorFlow, Numpy and pandas library to implement this.

Methodology

- Also plotted Precision recall curve and calculated various values like AUC, test accuracy, validation accuracy, cross entropy loss to study each model.
- We performed hyperparameter tuning to achieve the best accuracy in every model.
- We also tried ensemble methods, with multiple models including a combination of Logistic Regression and SVM, KNN and CNN.
- We found out the best accuracy method being CNN alone.

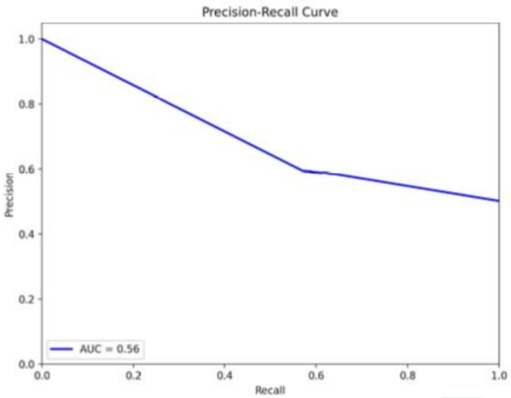
Logistic Regression

Value	Validation	Test
Accuracy	0.6793	0.67725
Precision	0.66386	0.6926
Recall	0.7218	0.6374
F1-Score	0.6916	0.6638
Specificity	0.6370	0.6374
Confusion Matrix:	[[6393 3642] [2772 7193]]	[[6374 3626] [2829 7171]]
False Positive Rate	0.3629	0.3626



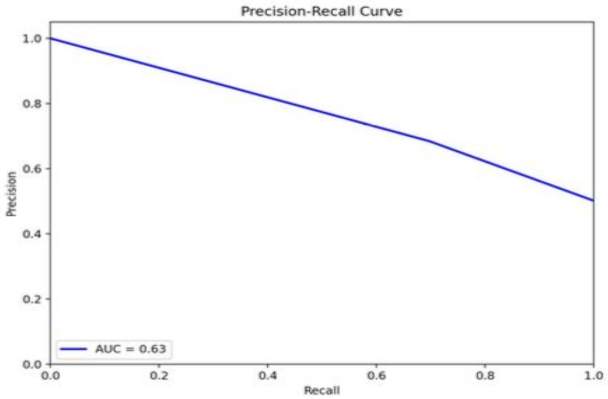
Naïve Bayes Classifier

Value	Validation	Test
Accuracy	0.5893	0.59275
Precision	0.59412	0.59570
Recall	0.57269	0.5773
F1-Score	0.58321	0.5863
Specificity	0.60602	0.6082
Confusion Matrix:	[[6039 3926] [4288 5747]]	[[6082 3918] [4227 5773]]
False Positive Rate	0.3939	0.3918



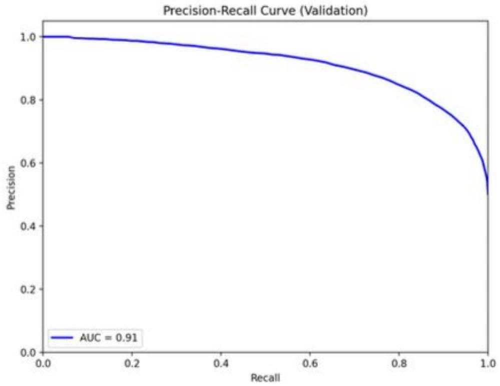
Decision Tree Classifier

Value	Validation	Test
Accuracy	0.6869	0.69745
Precision	0.68457	0.6983
Recall	0.6972	0.6953
F1-Score	0.6908	0.6967
Specificity	0.6764	0.6996
Confusion Matrix:	[[6741 3224] [3038 6997]]	[[6996 3004] [3047 6953]]
False Positive Rate	0.3235	0.3004



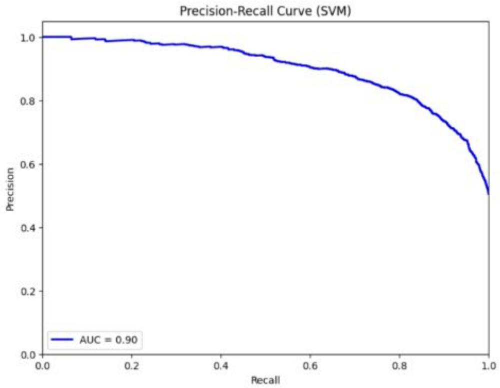
Random Forest Classifier

Value	Validation	Test
Accuracy	0.8272	0.82925
Precision	0.8498	0.80630
Recall	0.7963	0.8667
F1-Score	0.8222	0.8354
Specificity	0.8583	0.8667
Confusion Matrix:	[[8553 1412] [2044 7991]]	[[8667 1333] [2082 7918]]
False Positive Rate	0.1416	0.1333



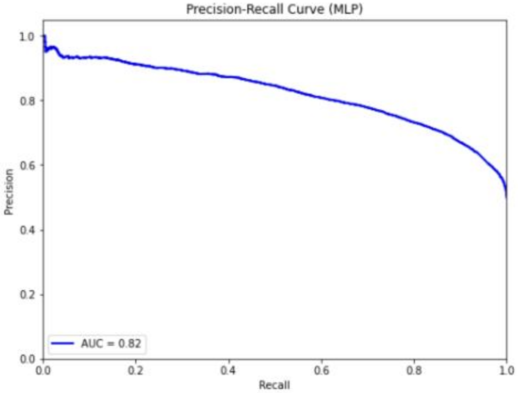
Support Vector Machine

Value	Validation	Test
Accuracy	0.814	0.8108
Precision	0.81357	0.8144
Recall	0.8192	0.805
F1-Score	0.8163	0.8096
Specificity	0.8086	0.8166
Confusion Matrix:	[[1602 379] [365 1654]]	[[8166 1834] [1950 8050]]
False Positive Rate	0.1913	0.1834



Multi Layer Perceptron

Value	Validation	Test
Accuracy	0.7543	0.7495
Precision	0.7328	0.7304
Recall	0.7976	0.7909
F1-Score	0.7638	0.7594
Specificity	0.7112	0.7081
Confusion Matrix:	[[7137 2898] [2016 7949]]	[[7081 2919] [2091 7909]]
False Positive Rate	0.288	0.2919



Convolution Neural Network

Our CNN model is designed for image classification, featuring three Convolutional layers with increasing filter depths (32, 64, 128) and ReLU activation. MaxPooling layers follow each convolution to downsample spatial dimensions. The model then flattens the output and connects to two Dense layers (128 units, ReLU, and 1 unit, sigmoid). It's compiled with Adam optimizer, binary cross-entropy loss, and accuracy metric, making it suitable for our binary image classification task.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 30, 30, 32)        896

 max_pooling2d (MaxPooling2  (None, 15, 15, 32)        0
 D)

 conv2d_1 (Conv2D)           (None, 13, 13, 64)        18496

 max_pooling2d_1 (MaxPoolin  (None, 6, 6, 64)          0
 g2D)

 conv2d_2 (Conv2D)           (None, 4, 4, 128)         73856

 flatten (Flatten)           (None, 2048)              0

 dense (Dense)               (None, 128)               262272

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 355649 (1.36 MB)
Trainable params: 355649 (1.36 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
Found 100000 images belonging to 2 classes.
Found 20000 images belonging to 2 classes.
Epoch 1/15
3125/3125 [==============================] - 30s 10ms/step - loss: 0.3232 - accuracy: 0.8586 - val_loss: 0.2223 - val_accuracy: 0.9101
Epoch 2/15
3125/3125 [==============================] - 28s 9ms/step - loss: 0.2196 - accuracy: 0.9112 - val_loss: 0.2050 - val_accuracy: 0.9151
Epoch 3/15
3125/3125 [==============================] - 28s 9ms/step - loss: 0.1900 - accuracy: 0.9254 - val_loss: 0.2010 - val_accuracy: 0.9190
Epoch 4/15
3125/3125 [==============================] - 28s 9ms/step - loss: 0.1693 - accuracy: 0.9330 - val_loss: 0.1698 - val_accuracy: 0.9327
Epoch 5/15
3125/3125 [==============================] - 27s 9ms/step - loss: 0.1520 - accuracy: 0.9409 - val_loss: 0.1709 - val_accuracy: 0.9334
Epoch 6/15
3125/3125 [==============================] - 27s 9ms/step - loss: 0.1373 - accuracy: 0.9469 - val_loss: 0.2293 - val_accuracy: 0.9133
Epoch 7/15
3125/3125 [==============================] - 27s 9ms/step - loss: 0.1253 - accuracy: 0.9517 - val_loss: 0.1754 - val_accuracy: 0.9330
```

Convolution Neural Network

Value	Validation	Test
Accuracy	0.9263	0.925
Precision	0.92022	0.9181
Recall	0.9330	0.9332
F1-Score	0.9155	0.9256
Specificity	0.8788	0.9168
Confusion Matrix:	[[8819 1216] [526 9439]]	[[9168 832] [668 9332]]
False Positive Rate	0.0803	0.0832

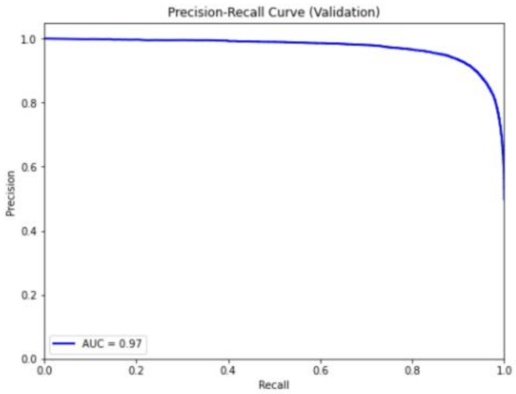


Figure 13. Precision Recall Curve CNN

Results and Analysis

- We evaluated seven machine learning models, and found varying levels of performance.
- CNN achieved the highest test accuracy, reaching 92.5%, making it the top-performing model in our study. It is due to its ability to automatically learn hierarchical features and spatial hierarchies, enabling it to capture intricate patterns and relationships within images.
- Naïve Bayes exhibited the poorest performance with a test accuracy of 59.275%, due to its simplistic assumption of feature independence.
- Random Forests, MLP and SVM also showed good accuracies ranging from 80% to 90%.

Results and Analysis

Model	Validation Accuracy	Test Accuracy
Logistic Regression	0.6793	0.67725
Naïve Bayes	0.5893	0.59275
Decision Tree	0.69345	0.69765
Multi Layer Perceptron	0.7543	0.7495
Support Vector Machine	0.814	0.8108
Random Forest	0.8266	0.8307
Convolution Neural Network	0.9263	0.925

Conclusions

- In conclusion, this project addresses the critical challenge of distinguishing real images from AI-generated ones, which have the potential to spread misinformation and manipulate public opinion.
- By utilizing the CIFAKE[2] dataset and employing various machine learning models, we have achieved promising results.
- The project has various limitations like hardware requirements for larger image datasets, and lesser research papers due to novelty of topic.

Individual Contributions

- All team members have equal contribution in the project.
- Data Visualization - All 3 members
- Data Preprocessing - All 3 members contributed and analysed
- Model Training
  - Logistic Regression, CNN - Sanya
  - Naive Bayes, SVM - Brinda
  - Decision Tree, Random Forest, MLP - Parisha
- Report and Presentation Writing - All 3 members

References

- Will Cukierski. (2013). CIFAR-10 - Object Recognition in Images. Kaggle. https://kaggle.com/competitions/cifar-10
- Bird, J.J., Lotfi, A. (2023). CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images. arXiv preprint arXiv:2303.14126.
- M. Zhu, H. Chen, Q. Yan, X. Huang, G. Lin, W. Li, Z. Tu, H. Hu, J. Hu, and Y. Wang, "Genimage: A million-scale benchmark for detecting ai-generated image," arXiv preprint arXiv:2306.08571, 2023.
- Haiwei Wu, Jiantao Zhou, Shile Zhang, "Generalizable Synthetic Image Detection via Language-guided Contrastive Learning", arXiv preprint arXiv:2305.13800, 2023.
- Krizhevsky, A. (2009) Learning Multiple Layers of Features from Tiny Images. Technical Report TR-2009, University of Toronto, Toronto. https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf