Predicting AI-Generated vs. Real Images

Abstract

With the increasing prevalence of AI-generated images, distinguishing between real and synthetic images has become a critical task. This study explores the preprocessing steps, data cleaning, and the implementation of machine learning models, specifically Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Convolutional Neural Networks (CNN), to classify images as either AI-generated or real.

Introduction

The rapid advancement of AI techniques has resulted in the generation of highly realistic images. While these innovations have various applications, they also pose challenges in identifying the authenticity of visual data. This paper presents a methodology for preprocessing data and implementing machine learning models to predict whether an image is AI-generated or real.

Methodology

**Data Preprocessing**

The dataset was cleaned and prepared to ensure high-quality inputs for the models. Key preprocessing steps included:

* **Data Cleaning**: Removal of corrupted and irrelevant files.
* **Feature Extraction**: Using feature engineering techniques, relevant image features were extracted to distinguish real and AI-generated images.
* **Normalization**: Image features were normalized to enhance the performance of the models.
* **Class Balancing**: To address class imbalance, oversampling techniques such as SMOTE were applied where necessary.

Support Vector Machine (SVM)

SVM was implemented as a linear classifier to separate the two classes. The key features of the model include:

* Kernel Selection: Linear and Radial Basis Function (RBF) kernels were tested.
* Hyperparameter Tuning: Grid search was used to optimize parameters like C and gamma.
* Cross-Validation: A 10-fold cross-validation was performed to evaluate model robustness.

K-Nearest Neighbors (KNN)

KNN was implemented to classify images based on similarity. The main considerations were:

* Distance Metric: Euclidean distance was used to compute similarities.
* K Selection: The optimal value for K was determined using cross-validation.
* Performance Evaluation: Accuracy, precision, recall, and F1-score were calculated to assess the model.

**Convolutional Neural Network (CNN) Implementation**

A CNN was implemented to classify images as real or AI-generated. The architecture and process are detailed below:

* **Dataset Preparation**: Images were resized to 128x128 pixels and normalized.
* **Class Distribution**: Balanced dataset ensured fair training and evaluation.
* **Data Augmentation**: Applied techniques like rotation, zoom, and brightness adjustments to enhance model generalization.

**CNN Architecture**

* Input layer with image size (128x128x3).
* Five convolutional layers with increasing filters (32, 64, 128, 256, 512) using ReLU activation, each followed by batch normalization, max pooling, and dropout for regularization.
* A fully connected dense layer with 512 neurons, followed by a dropout layer.
* Output layer with a sigmoid activation function for binary classification.

**Training Process**

* Optimizer: Adam with a learning rate of 0.0005.
* Loss Function: Binary cross-entropy.
* Metrics: Accuracy.
* Callbacks: ReduceLROnPlateau for adaptive learning rate adjustment and EarlyStopping to prevent overfitting.

Results

SVM Performance

KNN Performance