AI vs Real Portrait Classifier using fastai

Built by: Noa & Shahar— Course: Deep Learning (2025)

Github link

Problem Definition: Detecting AI-Generated Portraits

Goal - why did we choose this problem?

Our goal is to build a classifier that can tell whether a face image is real or AI-generated. This is a simple binary classification task.

Relevance in a Digital Age

- **Rise of Generative AI:** The rapid advancement and accessibility of tools like Stable Diffusion have led to a proliferation of highly realistic AI-generated images.
- **Risk of Deepfakes:** The potential for misuse, including the creation of deceptive deepfakes, necessitates effective detection mechanisms.
- **Detecting Manipulated Images:** The ability to identify synthetic content is crucial for media authenticity, combating misinformation, and digital forensics.

The classes

- AI generated portrait
- Real portrait

Real vs. AI: Can You Tell the Difference?

Real Human Portrait

AI-Generated Portrait

Data collection process

Sources

Kaggle.com - for both real and AI generated portraits

Steps

- Downloading data
- Standarized the file extensions to ensure consistency
- Made sure all of the photos had a white background dor uniformity

Dataset size

Each of the classes has 100 photos for a total of 200 photos

Dataset Statistics & Visual Examples

Our dataset was carefully constructed to provide a robust training and validation set for our classifier.

REAL Images

Approximately 100+ unique human portraits.

AI Images

Approximately 100+ synthetically generated portraits.

This balanced distribution aims to prevent model bias towards a specific class.

Train/Validation Split

The dataset was divided into an 80/20 split, with 80% for training the model and 20% reserved for validation to assess performance on unseen data.

Below is a visual representation of a batch of images from our dataset, showcasing the variety and quality of both real and AI-generated portraits.

Screenshot of `dls.show_batch()` illustrating the dataset.

Model Pipeline: Leveraging fastai with ResNet18

We selected **ResNet18** as our core architecture due to its efficiency and proven performance on image classification tasks, making it ideal for this project.

Architecture Choice: ResNet18

- **Fast:** Efficient computation for quicker iteration.
- **Good on Mid-Size Datasets:** Balances performance with dataset scale.
- **Beginner-Friendly:** Suitable for initial classifier development and prototyping.

Data Loading & Preprocessing

`ImageDataLoaders.from_folder` was used for efficient data loading, automatically handling labels based on folder names.

Images were resized to `224x224` pixels to fit ResNet18's input requirements and ensure consistent processing.

Model Fine-tuning

We employed the `fine_tune(3)` method, leveraging transfer learning by adapting a pre-trained ResNet18 model to our specific task over 3 epochs.

Performance Metrics

The primary metric for evaluating our model's performance was `error_rate`, providing a clear indication of misclassification frequency.

Training Results: Performance and Stability

The training process yielded promising results, demonstrating the model's ability to learn effectively from our dataset.

Learning curve showing loss decrease and accuracy increase over epochs.

Key Observations

- **Loss Decreases:** Both the training and validation loss consistently decreased over epochs, indicating that the model was successfully learning the underlying patterns.
- **Stable Training:** The learning curves show a stable training process without significant overfitting or underfitting, suggesting good hyperparameter tuning.
- **Good Generalization:** The validation loss closely tracked the training loss, implying the model generalized well to unseen data.

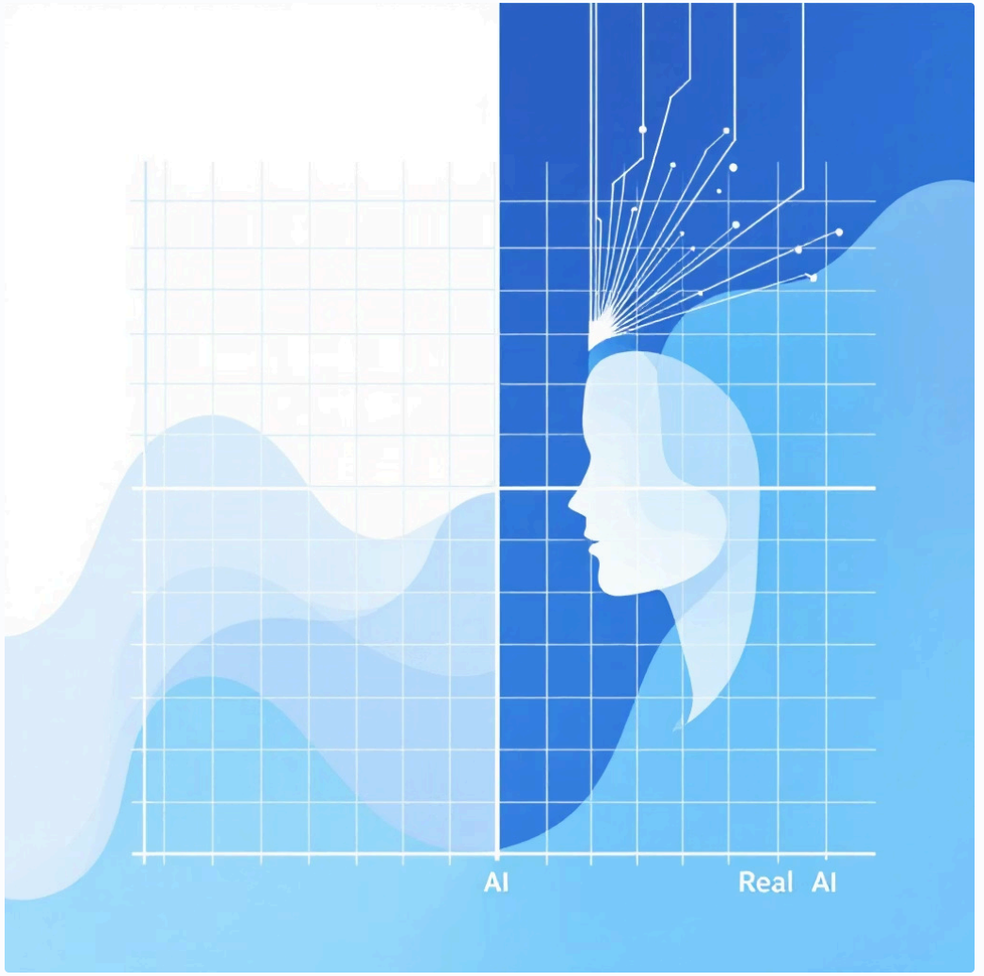
~90%

Final Accuracy

The model achieved a strong final accuracy on the validation set.

Confusion Matrix: Detailed Error Analysis

The confusion matrix provides a granular view of our classifier's performance, highlighting where the model succeeded and where it struggled.



Screenshot of `interp.plot_confusion_matrix()`.

Analyzing Misclassifications

- **Class with More Errors:** Based on the matrix, the model had a higher number of misclassifications when predicting real images as AI images.
- **Harder to Classify:** This suggests that real portraits are inherently harder for the model to correctly identify. This could be due to subtle artifacts, variations in generation styles, or a higher degree of photorealism that blurs the lines with AI photos. Further analysis will focus on these areas.

Challenged encountered

- **High Visual Similarity** - Modern generative models produce extremely realistic faces.
- **Limited Dataset Size** - Collecting diverse and balanced datasets for both classes is challenging.
- **Dataset Bias Between Classes** - AI-generated images often have clean, uniform backgrounds and perfect lighting, while real photos contain varied environments, lighting conditions, and noise.
- **Ethical & Robustness Issues** - A model trained today might fail on next-generation synthetic faces.