We work for tomorrow

UNI GRAZ

MICRO-DEGREE

AI & Society

Module C Practical Implementation

PAPER TOPIC

From Six Numbers to Millions of Weights

An Exploratory Study in AI-Generated Image Detection

AUTHORS

Samah Gourari

Davit Neberidze

Klimova Sophia Theodora

Mahboubeh Najafzadeh

Alisa Baralic

Supervised by Prof. Di Natale Anna

University of Graz ▪ IDea Lab

August 3, 2026

Abstract

AI image generators have made photorealistic images easy to produce, and detecting them is no longer a matter of spotting a human editor's mistakes but of finding the statistical fingerprints a machine leaves behind. This project, carried out as part of the AI & Society Micro-Degree at the University of Graz, explores AI-generated image detection from first principles instead of aiming for a perfect solution. We built and compared three detectors of increasing complexity. A Random Forest using six handcrafted image statistics, a Random Forest using raw pixel values, and a fine-tuned EfficientNet-V2-S convolutional network. All three achieved strong accuracy on their training dataset (up to 93% F1 for EfficientNet), but each failed when tested on images from outside that dataset, including a real personal photograph misclassified as AI-generated with 65.67% confidence. Rather than treating this as simple overfitting, we traced the failure to the training data itself, the real images were polished stock photography, and the fake images came from a narrow set of generators, so the models had learned to distinguish a specific visual style rather than a general concept of "real versus AI-generated." Retraining on a more diverse and naturalistic dataset improved generalisation (raising Raw Pixels accuracy on unseen data from 53.8% to 71.5%), though it remained far from reliable. Overall it was two independently developed approaches, one was interpretable and handcrafted and one was deep and opaque, both converged on the same diagnosis, which is that the composition of the dataset matters more than the complexity of the model. This exploratory study does not solve AI-generated image detection, but it offers a transparent account of why the problem is harder than it first appears, and where future efforts should focus.

Contents

1 Introduction

Before AI image generators existed, fake images were already a problem, but in a different kind. Doctored photographs, spliced faces, cloned backgrounds. The tools were Photoshop and patience. Detection back then was about catching the seams: inconsistent lighting, mismatched shadows, JPEG artifacts from repeated saves, cloned regions that repeated pixel patterns the human eye could eventually spot if it knew what to look for. It was a game of human mistakes, and detectors were built to find those mistakes.

Then came AI generators. Midjourney, DALL-E, Stable Diffusion. One prompt and a few seconds later, you have a photorealistic image of something that never existed, a person or a place. The reason these generators work so well is because they are trained on billions of real images from the internet, and from that data they learn to imitate what reality looks like with remarkable accuracy. The more data, the better the imitation. And we live in a world that keeps producing more data every day.

This changes the core of the detection. You are no longer looking for a human's editing mistakes. You are looking for the statistical fingerprints of a machine that was specifically trained to produce images that look real. Those fingerprints vary between generators.

Artificial intelligence, as understood throughout the AI & Society Micro-Degree at the University of Graz, is the ability of a computer system to perform tasks that would normally require human intelligence, by learning patterns from large amounts of data rather than following explicitly programmed rules. In the context of image generation, this means a model that has seen so many real photographs that it learned, statistically, what a real image looks like and how to reproduce that appearance from scratch.

So how do we detect what such a model produces? That is the question this paper tries to answer, not by presenting a state-of-the-art solution, but by building the understanding from scratch. What does AI leave behind in an image? How do simple visual features compare to deep learning? When does detection work, and more importantly, when does it fail?

This project was carried out as part of the AI & Society Micro-Degree at the University of Graz, supervised by professor Di Natale Anna from the University of Graz. The goal was not to build the best model but it was to understand the problem honestly and share that understanding in a way that is useful to anyone curious enough to ask the same questions we did.

2 What Makes an Image AI-Generated, and Why Does It Matter?

2.1 How Does an AI Actually Generate an Image?

To make it intuitive, imagine you want to generate an image of a cat. You do not draw one. You do not copy one. Instead, you feed a model millions of photographs of cats from across the internet, cats of different breeds, colors, sizes, lighting conditions, backgrounds. Every one of those images is, at its core, just numbers. A pixel is a combination of three values: red, green, and blue, each between 0 and 255. An image is nothing more than a grid of those combinations. The computer never sees a cat. It sees numbers.

From those numbers, the model learns patterns. Not what a cat is, but what a cat statistically looks like. What color values tend to appear together. What shapes are common. What textures repeat. When you then ask it to generate a new cat image, it does not retrieve a cat from memory. It constructs one, pixel by pixel, as a statistically plausible combination of everything it has seen [4]. The result tends to look like the most "average" cat imaginable, because it is, in a sense, the average of millions of cats compressed into a set of learned numerical patterns.

This is why AI-generated images can look so convincing. The model is not guessing. It is reconstructing reality from its statistical understanding of it. And this is also why detection is hard because there is no obvious mistake to catch and no clumsy edit to spot. The image was built correctly, just not by a camera.

What the model does leave behind is subtler. Statistical regularities that do not quite match how a real camera captures light. Textures that are slightly too smooth or too repetitive. Frequency patterns in the pixel grid that differ from natural image noise. These are the fingerprints we are trying to detect.

2.2 Why This Matters: What We Learned at Uni Graz

The AI & Society Micro-Degree at the University of Graz did not just teach us how AI works technically. It asked us to think about what AI does to the world around us. Image generation is a good example of why that question matters, because the same technology that can produce a beautiful piece of art can also produce a fake photograph of a politician doing something they never did.

Ethically, the question is genuinely hard. Generating an image of a fictional cat is harmless. Generating a photorealistic image of a real person in a situation that never happened is something else completely. There is no universal answer to where the line is, it means that the ethics of AI image generation cannot be solved by technology alone.

They require judgment, context, and ongoing debate [5].

Legally, the European Union has started providing some structure through the AI Act, which is one of the first serious attempts to regulate AI systems by risk level [3]. AI-generated images that could deceive the public, influence elections, or violate someone's likeness fall into high-risk categories that require transparency and accountability. But legislation moves slowly, and the technology does not. Beyond questions of risk and deception, legal scholars have also pointed out that AI-generated content raises unresolved questions of authorship and moral rights, since existing copyright frameworks were not built with machine-generated works in mind [8].

Educationally, the spread of convincing fake images affects how people learn and what they trust. When students, journalists, or the general public can no longer reliably tell what is real, the foundations of informed decision-making start to erode [6]. Detection tools are part of the answer, but so is media literacy.

Economically, AI image generation puts pressure on photographers, illustrators, and creative professionals whose work can now be imitated or replaced at near-zero cost. At the same time, it opens new possibilities for people who could not previously afford visual content. The economic impact cuts both ways, and it is still unfolding.

None of these questions have clean answers. But they are the reason why understanding how detection works, even at a basic level, matters beyond the technical. If we cannot tell what is real, all of the above becomes harder to navigate.

3 Our Approach: From an Image to a Decision

3.1 What Is Machine Learning, and What Did We Use?

Machine learning is a way of letting a computer find patterns in data without being explicitly told what those patterns are. Instead of writing rules by hand, like "if the image has blurry edges, it is fake", the computer is given many labelled examples and tasked with figuring out the rules on its own. The quality of what it learns depends on the data it sees, and the method it uses.

For this project, we used two methods that represent very different levels of complexity: a Random Forest, and a deep learning model called EfficientNet-V2-S. Before explaining either one, it is worth understanding what a machine learning model actually is.

3.2 What Is a Model?

A machine learning model is, at its core, a function. It takes some input, in our case numbers representing an image, and produces an output: a decision. Real or fake.

The word "model" just means a mathematical object that maps inputs to outputs. The difference between models is in how that mapping is constructed and what form it takes. A Random Forest constructs it as a collection of decision trees. A neural network constructs it as a chain of mathematical operations inspired loosely by how biological neurons work. The goal in both cases is to find the mapping that produces the correct output as often as possible on new, unseen data.

3.3 What Does Training Mean?

Training is the process of building that mapping from data.

The data is split into two parts before training begins, 80% for training and 20% for testing. The model never sees the test set during training. It uses only the 80% to find patterns. The 20% is kept aside and used at the end to evaluate how well the model performs on examples it has never encountered. This separation is essential.

What happens during training depends on the type of model. But in general, the model is not memorising examples. It is extracting structure from them. A model that memorises will fail completely on new data. A model that has genuinely extracted a pattern will perform reasonably on data that follows the same distribution, even if it has never seen those specific examples before.

The question of whether what the model learned is truly general, or just specific to the training data, is the main question of this entire study.

3.4 Random Forest

A decision tree classifies an input by asking a sequence of yes or no questions. Is the brightness above 120? If yes, is the contrast below 30? If yes, predict fake. If no, predict real. And so on, until the tree reaches a final answer. Each question splits the data at a threshold on one feature, and the threshold is chosen during training to be the one that best separates real from fake images at that point.

A Random Forest trains many of these trees, each on a slightly different random sample of the training data and using a random subset of features at each split [1]. When a new image arrives, all trees produce a prediction independently, and the majority vote wins.

What the model stores in memory is not images but the structure of the trees. At each node, which feature to check, what threshold to compare against, and which branch to follow. For our simple features model, this means checking one of six numbers (brightness, contrast, sharpness, or a colour channel average) against a threshold value. That is the entirety of the model's knowledge. When the session closes, it is gone from memory unless saved to a file.

Random Forest does not require the input data to be normalised, it trains fast on small datasets, and it is interpretable, we can inspect which features the trees use most and where they set their thresholds.

subsectionNeural Networks: From One Neuron to EfficientNet

To understand EfficientNet, it helps to start from the very beginning: a single artificial neuron.

A neuron receives some inputs, multiplies each one by a number called a **weight**, and sums the results. That sum then passes through an **activation function**, which introduces non-linearity. Without this step, stacking many neurons together would still just produce a linear function, which severely limits what the network can learn.

The output of that one neuron is then passed as input to other neurons in the next layer. Stack many neurons in many layers, and the network starts to compose increasingly abstract transformations of the input. Early layers detect simple patterns like edges and colour gradients. Later layers combine those into textures, shapes, and eventually higher-level concepts. This is called a **neural network**.

So what do we actually feed a neural network, and how is that different from what we gave the Random Forest? For the Random Forest, we designed the features ourselves like brightness, contrast, sharpness, colour averages, numbers we chose because we believed they might separate real images from fake ones. A neural network like EfficientNet-V2-S is fed the pixels of the image directly, with no summarising in between. A 224×224 image has three colour values per pixel, red, green, and blue, so the network receives roughly 150,000 numbers per image and nothing else. Instead of telling the network "here is the contrast, here is the sharpness," we are basically saying "here are all the raw numbers, figure out for yourself what matters." This is closer to how a person recognises a cat, we do not consciously calculate its contrast or colour averages, we notice edges, curves, eyes, fur, without thinking about the pixel values at all. A neural network is trying to arrive at a similar kind of recognition, except it has to learn, entirely from data, which patterns in those raw pixel values are worth paying attention to in the first place.

During training, the network makes a prediction for each training example, compares that prediction to the correct answer, and calculates an error, called the **loss**. The goal is to minimise that loss. To do this, the network uses **backpropagation**, it traces back through the layers and calculates how much each weight contributed to the error [9]. Then it adjusts each weight by a small amount in the direction that reduces the error. The size of each adjustment is controlled by the **learning rate**. This process repeats over many examples and many passes through the data (called epochs) until the loss is as small as the network can get it. The update rule for each weight is:

$$w_{\text{new}} = w_{\text{old}} - \eta \cdot \frac{\partial L}{\partial w}$$

where η is the learning rate and $\frac{\partial L}{\partial w}$ is the gradient of the loss with respect to that weight, telling the network in which direction the weight should move, which is called **gradient descent**.

What the network stores is millions of weight values, one for each connection between neurons, adjusted through training to minimise prediction error. Because these are just numbers that get nudged incrementally, it is possible to continue training a network on new data without starting from scratch. This is called **fine-tuning**, and it is one of the key practical differences from a Random Forest, which has no equivalent mechanism, its tree structure is built once and cannot be updated.

EfficientNet-V2-S is a convolutional neural network, meaning its early layers use filters that slide across the image and detect local patterns [10]. It was originally trained on ImageNet, a dataset of over a million images across 1,000 categories, giving it a rich initial understanding of visual patterns before it ever sees a fake image [2]. We replaced its final classification layer with a two-class output (real or fake) and fine-tuned the entire network on 43,000 images from the Kaggle dataset, using the AdamW optimiser (the algorithm that controls how the weights are updated at each step, adjusting the learning rate automatically for each weight rather than using one fixed step size for all of them) over 12 training epochs [7]. Data augmentation was applied during training: random crops (randomly cutting out a portion of the image so the model does not always see the full frame), colour jitter (randomly shifting the brightness, contrast, and saturation so the model is not fooled by lighting differences), and Gaussian blur (slightly blurring the image to simulate lower-quality captures), alongside JPEG compression at varying quality levels, all designed to expose the model to variation it might encounter in real-world conditions.

Table 1: Key differences between Random Forest and EfficientNet

Property	**Random Forest**	**EfficientNet**
What it stores	Fixed decision rules	Adjustable weight values
Can update with new data	Full retrain only	Fine-tune existing weights
Input	6 or 12,288 numbers	Full image
Interpretability	High	Low
Training time	Fast	Slow (GPU required)

3.5 What Is an Image to a Computer?

A pixel is three numbers: one for red, one for green, one for blue, each between 0 and 255. An image is a grid of pixels. A 128×128 image is $128 \times 128 \times 3 = 49{,}152$ numbers.

That is all that enters any model in this study. Python libraries read the image file and convert it into a numerical array.

3.6 Simple Features vs. Raw Pixels

Given that an image is just numbers, the question is which numbers to give the model.

Simple features reduce each image to six summary statistics brightness (average pixel value across the grayscale version), contrast (standard deviation of pixel values), sharpness (average difference between horizontally adjacent pixels), and average red, green, and blue. Six numbers per image. Almost all visual information is discarded. The model only works with rough global statistics, nothing about local structure or texture.

Raw pixels keep everything. Each image is resized to 64×64, flattened into 12,288 numbers, and normalised to a 0–1 range. The model sees the full image, pixel by pixel, with access to fine-grained texture and local patterns. Much more information, but also much more risk of the model learning patterns that are specific to the training data and do not generalise.

The full Kaggle dataset has 60,000 images. Training a Random Forest on 60,000 images with 12,288 features each would be computationally slow. For an exploratory study, a smaller controlled subset makes experiments faster, reproducible, and easier to reason about. We therefore sampled 500 real and 500 fake images, with a fixed seed (42) to ensure reproducibility. This gives 1,000 images: 800 for training, 200 for testing.

3.7 Testing Robustness with the Degraded Dataset

After testing on clean images, we wanted to see what happens when image quality drops. The motivation is that social media platforms compress images automatically when users upload them, destroying pixel-level detail. If a detection model relies on subtle pixel patterns left by AI generators, compression could remove exactly the evidence it needs.

We simulated this with three steps applied to all 1,000 images: shrink to 32×32 pixels, blow back up to 128×128 (producing blur and pixelation), and save as a JPEG at quality 20 out of 100. A typical good-quality JPEG is saved at 80 to 95. We then extracted features and pixels from these degraded images using the same functions as before, and evaluated the same trained models without retraining anything.

3.8 A First External Test

Both experiments above used images from the same Kaggle dataset the models were trained on. To get a first signal of whether the models learned something general or

something dataset-specific, we tested them on two completely external images, one real personal photograph and one AI-generated image made with Microsoft Designer.

Two images is not a meaningful sample, and we are not claiming any conclusions from it. It is simply a first look. But as a directional signal, it pointed toward a pattern that the later, larger experiments confirmed.

3.9 EfficientNet and a Revised Diagnosis

Before reading the numbers, it helps to clarify three terms. An **epoch** is one complete pass through the entire training dataset. After each epoch, the model has seen every training example once and has updated its weights accordingly. **Accuracy** is the simplest performance measure, out of all predictions made, how many were correct. **F1 score** is a more balanced measure that accounts for both false positives (predicting fake when the image is real) and false negatives (predicting real when the image is fake). When the two classes are roughly equal in size, as they are here, F1 and accuracy tend to tell a similar story.

EfficientNet model achieved 93% F1 on the Kaggle evaluation set after 12 epochs. The F1 score rose from 78% at epoch 1 to 93% by epoch 12, showing the network gradually adjusting its weights to better separate real from fake images. When tested on a personal photograph outside the dataset, however, it predicted "AI-Generated" at 65.67% confidence on a real image. It was wrong.

After revising the interpretation. The 93% score was likely inflated because the evaluation set contained AI images from the same generators the model trained on. The test was not as hard as it appeared. The model had learned to recognise the visual style of specific generators, not a general concept of what AI-generated means. This is a dataset problem as much as a model problem, and it became the central question driving the second round of experiments.

3.10 The Second Round: Testing Generalisation

After the first round, the picture was clear, both approaches struggled to generalise beyond the data they were trained on. The likely reason is the dataset itself. The real images in the Kaggle set came from photography and art platforms, which tend to look polished and idealised. The fake images came from a specific set of generators. A model trained on this combination risks learning the difference between "idealised stock photography" and "this particular generator's style", instead of the difference between real and AI-generated in any general sense.

To investigate this, we introduced a second dataset: the Shutterstock AI vs Human-

Generated Image dataset, which contains more naturalistic real photographs. We sampled 500 real and 500 fake images from it using the same procedure as before.

We then ran three tests. First, we applied the original Kaggle-trained models directly to the new dataset without retraining, to see whether what they learned transfers at all. Second, we combined both datasets and trained fresh models from scratch on the combined data, to see whether more diverse training produces a more general model. Third, we extended the external test, instead of two images, we used 12 personal images (7 real, 5 AI-generated, labelled manually) and tested all four models on them. We also ran all four models on 6 image pairs, each pair consisted of the same image before and after real social media compression, to see whether actual platform compression changes the model's predictions.

The results of all these experiments are presented in the next section.

4 Results

This section presents what the experiments described above actually showed. As stated throughout, none of these numbers should be read as final conclusions. Given the small subsets, single external test images, and exploratory nature of the study, the goal is to observe patterns honestly, not to prove a hypothesis.

4.1 Results on the Original (Kaggle) Dataset

The first question was simply: on the dataset both methods were trained on, how well do they do? Table 2 shows accuracy for both methods, on both the clean subset and the degraded (blurred, low-quality JPEG) version of the same 1,000 images.

Table 2: Random Forest accuracy on the Kaggle subset, clean vs. degraded

Method	**Clean**	**Degraded**
Simple Features (6 numbers)	49.5%	52.5%
Raw Pixels (12,288 numbers)	60.5%	62.0%

A few things stand out. First, on clean images, Raw Pixels clearly outperforms Simple Features (60.5% vs. 49.5%), which makes sense, the model has access to far more information about the image. Second, and more surprising, both methods performed slightly better on the degraded images than on the clean ones. This runs counter to the intuition that damaging an image would make detection harder. We are not confident enough in this result to explain it definitively, given the subset size, but one plausible reading is that

the specific degradation process (shrinking, blurring, and re-compressing) may have introduced its own consistent artifacts that happened to correlate with the fake/real labels in this particular subset, rather than genuinely making the task harder.

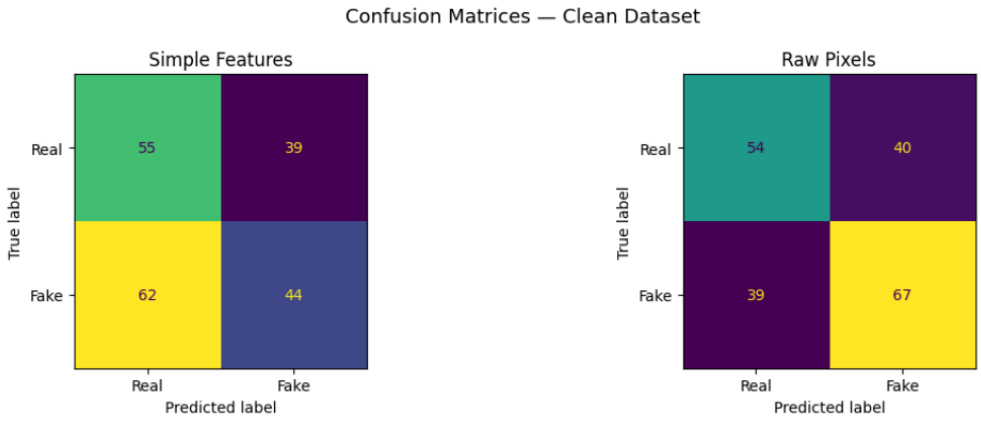

Figure 1: Confusion matrices for Simple Features and Raw Pixels on the clean Kaggle subset (200 test images).

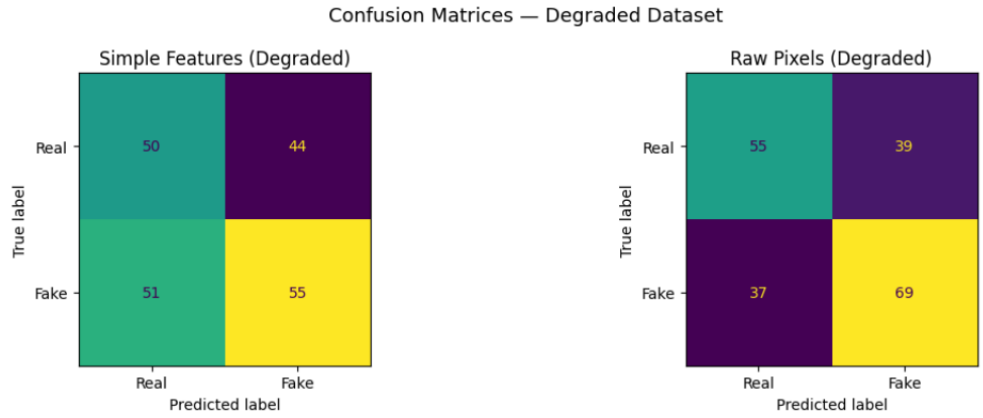

Figure 2: Confusion matrices for Simple Features and Raw Pixels on the degraded version of the same subset.

To understand why the simple features behave the way they do, it helps to look at how brightness, contrast, and sharpness are actually distributed between real and fake images in the clean subset (Figure 3). The real and fake distributions overlap heavily for all three features, which is consistent with the relatively modest accuracy of the Simple Features model: if brightness, contrast, and sharpness do not separate real from fake very cleanly on their own, a model using only those six numbers cannot be expected to separate them cleanly either.

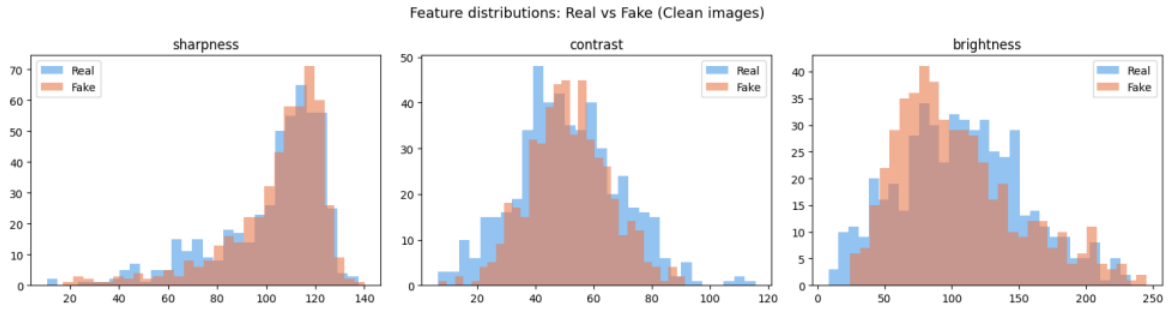

Figure 3: Distribution of sharpness, contrast, and brightness for real vs. fake images in the clean subset. Substantial overlap between the two classes for all three features suggests why simple statistical features alone struggle to separate real from fake reliably.

4.2 A First External Test

The Kaggle subset, clean or degraded, is still the same dataset the models were trained on. A better question is what happens with images the models have never seen in any form, images from outside the Kaggle collection entirely. As a first, small-scale look at this, we tested both models on two images, one real personal photograph, and one AI-generated image made with Microsoft Designer.

The Simple Features model classified both correctly. The Raw Pixels model misclassified the real photograph as fake, correctly identifying only the AI-generated image, for 50% accuracy on this tiny external set. Two images is far too small a sample to draw any real conclusion from, but it was the first hint of a pattern that the wider tests later in this section would reinforce. The model with access to more detailed information (raw pixels) was not necessarily the model that generalised better to images outside its training data.

4.3 EfficientNet-V2-S Results and a Revised Diagnosis

In parallel with the Random Forest experiments, we trained a deep learning model, EfficientNet-V2-S (described in Section 3.4), on the full 43,000-image Kaggle dataset.

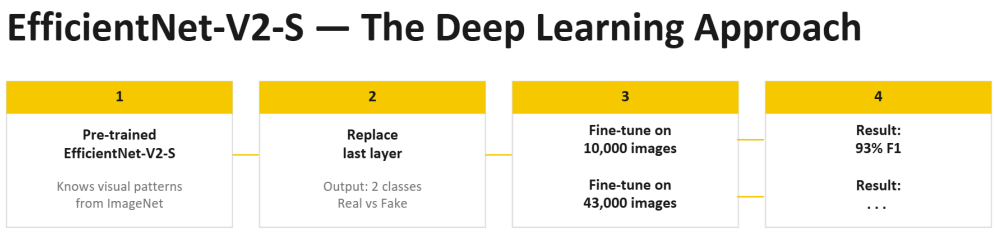

Figure 4: Overview of the EfficientNet-V2-S fine-tuning pipeline: starting from ImageNet pretraining, replacing the final layer with a two-class output, and fine-tuning on progressively larger subsets of the Kaggle dataset.

After 12 training epochs, the model reached 93% F1 on the Kaggle evaluation set, with 92.5% validation accuracy. Figure 5 summarises these headline numbers alongside the same generalisation failure we saw in our own external test, when applied to a personal photograph outside the training distribution, the model predicted "AI-Generated" with only 65.67% confidence, and it was wrong.

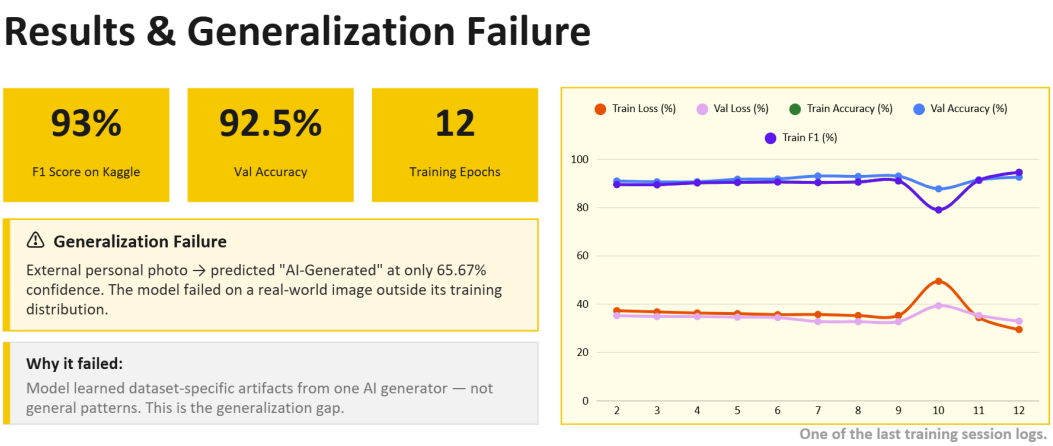

Figure 5: Headline results for EfficientNet-V2-S on the Kaggle evaluation set, alongside the confidence score on a misclassified external image.

Figure 6 shows the logged training and validation metrics across the fine-tuning process. Because the training script overwrote previous session logs rather than appending to them, this record is a composite of at least two separate training sessions rather than one uninterrupted run. The visible discontinuity around epoch 10, a spike in training loss and a dip in training F1, coincides with the learning rate resetting to its initial maximum value, which is consistent with the start of a new session rather than an instability occurring mid-training. The final epochs recover to metrics consistent with the reported 93% F1 and 92.5% validation accuracy, so the headline numbers are not in question, only the shape of the curve connecting them.

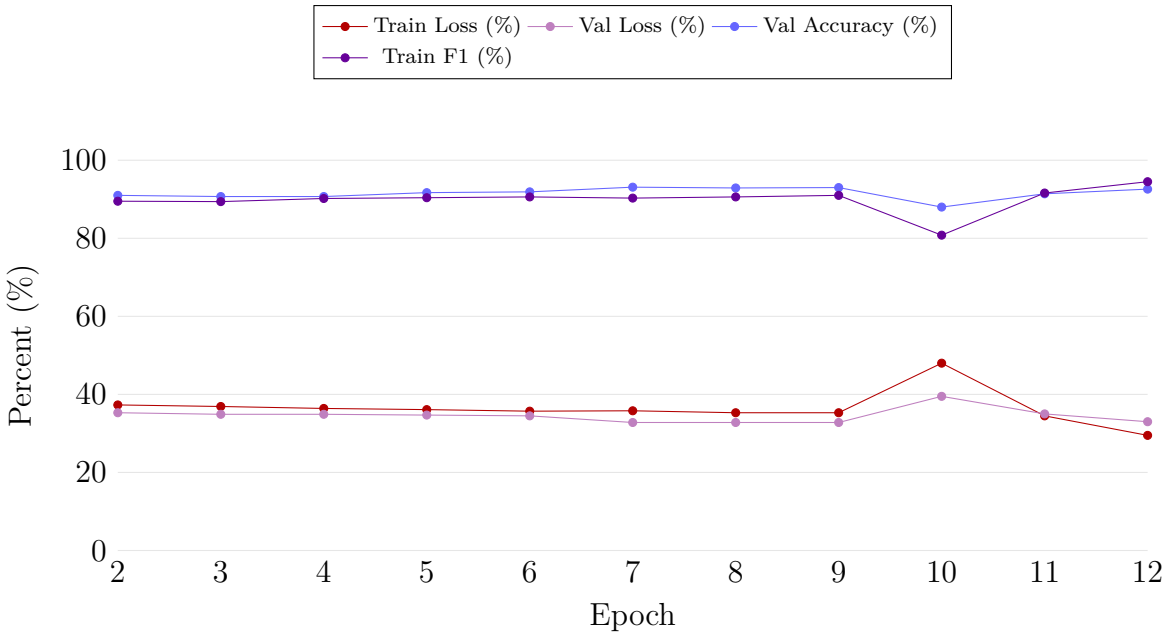

Figure 6: Training and validation metrics logged during EfficientNet-V2-S fine-tuning, reconstructed from the raw training logs (Epochs 2–12). Due to a logging limitation, this record is a composite of at least two separate training sessions rather than a single continuous run; the discontinuity at epoch 10, where training loss spikes and both training accuracy and F1 drop sharply, coincides with the learning rate resetting to its initial maximum value, consistent with the start of a new training session rather than an in-training instability. The final two epochs recover to metrics consistent with the reported 93% F1 and 92.5% validation accuracy.

As with our own Random Forest results, the initial interpretation of this generalisation failure was overfitting because the model had presumably learned dataset-specific artifacts from one AI generator rather than a general concept of "AI-generated". After integrating a new, more naturalistic real-image dataset (discussed below), this interpretation was revised. Figure 7 summarises both the original and the revised interpretation side by side.

Previously: A Generalisation Gap

93% F1 on the Kaggle evaluation set
92.5% validation accuracy
65.67% confidence on a misclassified personal photo

Old interpretation: the model was assumed to have overfit dataset-specific artifacts from one AI generator, rather than learning a general distinction between real and AI-generated images.

Now: A Revised Diagnosis

Effect of the new real-image dataset: incorporating a more realistic, unpolished real-image dataset changed the interpretation. Overfitting was found to be only partially true at most.

The real issue: the original F1 evaluation set was too easy. Its AI-generated images came from outdated generators, which likely inflated the score and made performance look better than it actually was.

Current focus: collecting newer AI-generated images from modern generators, combined with the new real-image dataset, to obtain a more realistic evaluation.

Figure 7: Revision of the generalisation-failure interpretation following integration of a more realistic real-image dataset. The original diagnosis (left) attributed poor generalisation primarily to overfitting on generator-specific artifacts; the revised diagnosis (right) identifies evaluation-set difficulty as a substantial contributing factor.

Concretely, this meant two changes to the dataset. On the real-image side, studio-style stock photography was supplemented with images from COCO (Common Objects in Context), a dataset of everyday, unposed scenes, to better simulate what an ordinary "camera roll" photo actually looks like. On the AI-generated side, images from older, flawed generators were filtered out in favour of images from more modern tools (Midjourney, Stable Diffusion) that had already solved the obvious visual artifacts, like distorted hands or warped backgrounds, forcing any detector to rely on subtler statistical signatures instead.

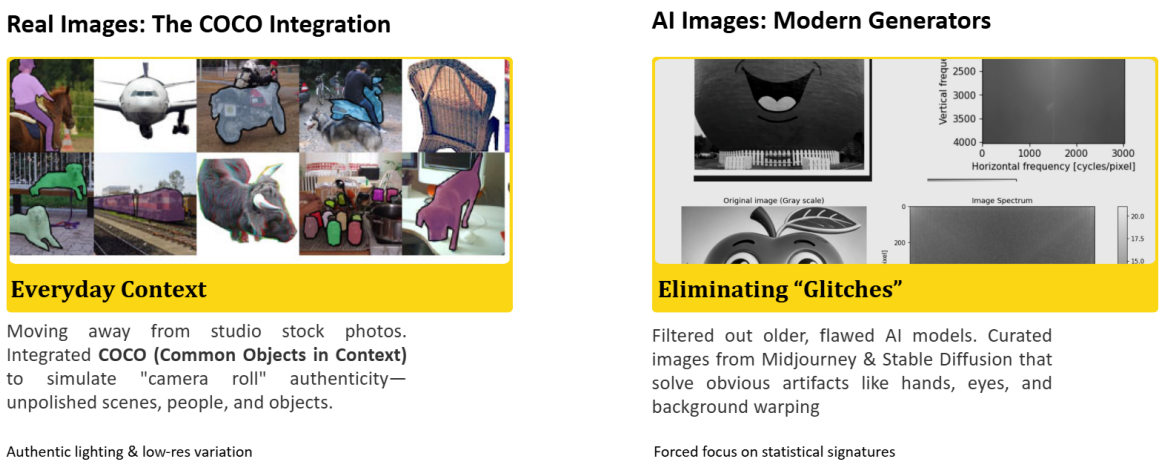

Figure 8: Dataset curation changes made in response to the revised diagnosis: integrating COCO for more naturalistic real images (left), and curating AI-generated images from modern generators that eliminate obvious visual artifacts (right).

This mirrors, independently, the same reasoning that motivated our own second round of experiments below: if, the real images in a dataset all look unusually polished, a model may learn to distinguish "polished stock photography" from "a specific generator's style", rather than learning anything general about what makes an image AI-generated.

4.4 The Second Round: Testing Generalisation

Motivated by the same concern, we introduced a second, more naturalistic dataset (Shutterstock AI vs. Human-Generated Images) and ran three further tests on the Random Forest models.

4.4.1 Test B (Old Models on the New Dataset, No Retraining)

First, we took the models exactly as trained on the Kaggle subset and applied them directly to 1,000 images from the new dataset, with no retraining at all. This tells us whether whatever the models learned transfers to a different, more realistic collection of real and fake images.

Table 3: Old (Kaggle-trained) models applied directly to the new dataset

Method	Accuracy
Simple Features	47.6%
Raw Pixels	53.8%

Both numbers are close to the 50% mark expected from random guessing on a balanced two-class problem. In other words, whatever the models had learned from the Kaggle subset barely transferred to the new, more naturalistic dataset at all.

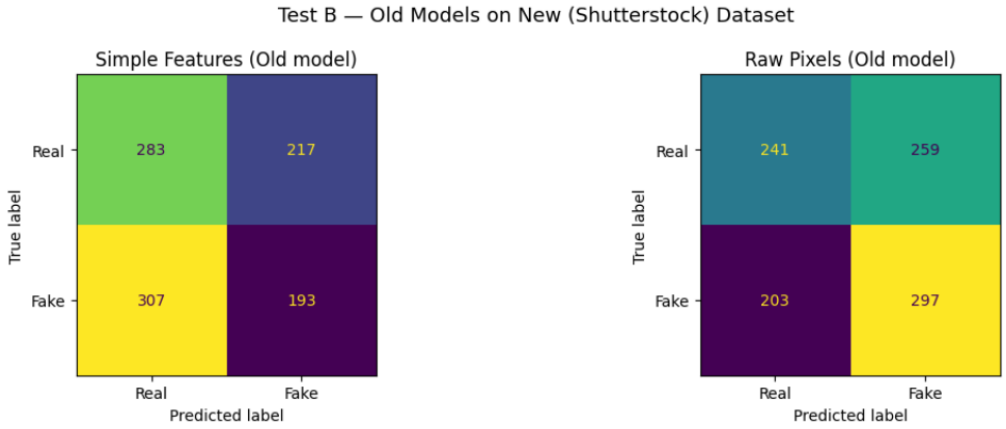

Figure 9: Confusion matrices for the old (Kaggle-trained) models, evaluated without retraining on the new (Shutterstock) dataset.

4.4.2 Test C (Fresh Models on the Combined Dataset)

Next, we trained entirely new Random Forest models from scratch on a combination of both datasets (the original Kaggle subset and the new Shutterstock subset), to see whether more diverse training data, covering both idealised and naturalistic real images, produces a model that generalises better.

Table 4: Fresh models trained on the combined (Kaggle + Shutterstock) dataset

Method	**Accuracy**
Simple Features	63.5%
Raw Pixels	71.5%

Both methods improved substantially compared to Test B, and Raw Pixels in particular reached its highest accuracy of any experiment in this study. This is consistent with the idea that dataset diversity, rather than model complexity alone, has a meaningful effect on how well a model generalises.

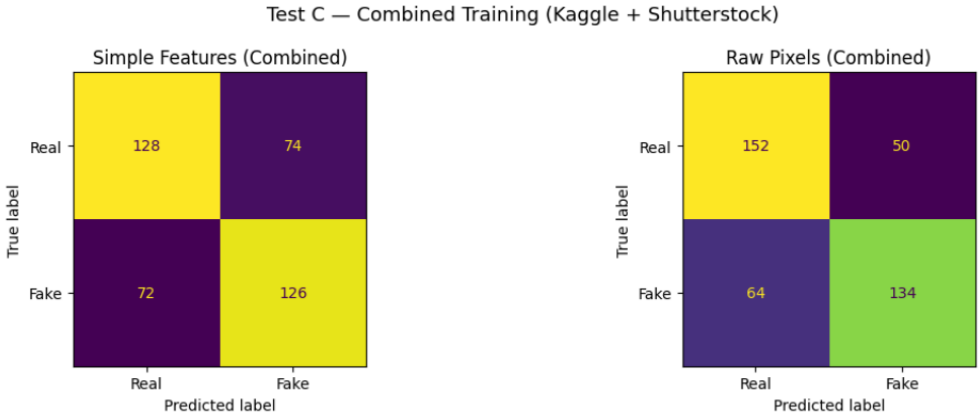

Figure 10: Confusion matrices for models freshly trained on the combined Kaggle and Shutterstock dataset.

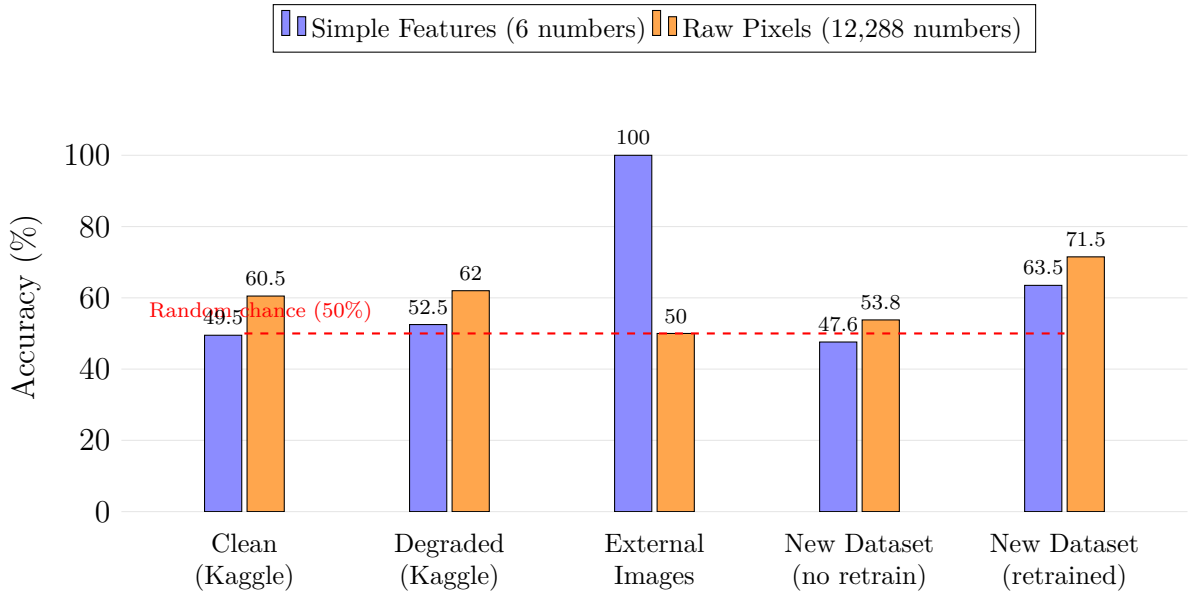

Figure 11: Accuracy of Simple Features and Raw Pixels across all Random Forest experiments: on the clean and degraded Kaggle subset, on the first two-image external test, on the new (Shutterstock) dataset without retraining, and on the new dataset after retraining on the combined data. Values match those reported in Tables 2, 3, and 4.

4.4.3 Test E (An Extended External Test with 12 Personal Images)

Two external images, as noted earlier, is not a meaningful sample. To get a more reliable signal, we extended the external test to 12 personal images (7 real, 5 AI-generated, labelled manually) and evaluated all four models on them: the two original Kaggle-trained models and the two freshly trained combined models.

Table 5: Accuracy on 12 personal images (Test E)

Model	**Accuracy**
Simple Features (Old, Kaggle-only)	66.7% (8/12)
Raw Pixels (Old, Kaggle-only)	75.0% (9/12)
Simple Features (Combined)	75.0% (9/12)
Raw Pixels (Combined)	75.0% (9/12)

In this larger and more personal set of external images, Raw Pixels performed at least as well as Simple Features across the board, and combining datasets brought the Simple Features model up to parity with both Raw Pixels models. This is a different picture than the very first two-image external test, another reminder of how much a tiny sample size can mislead.

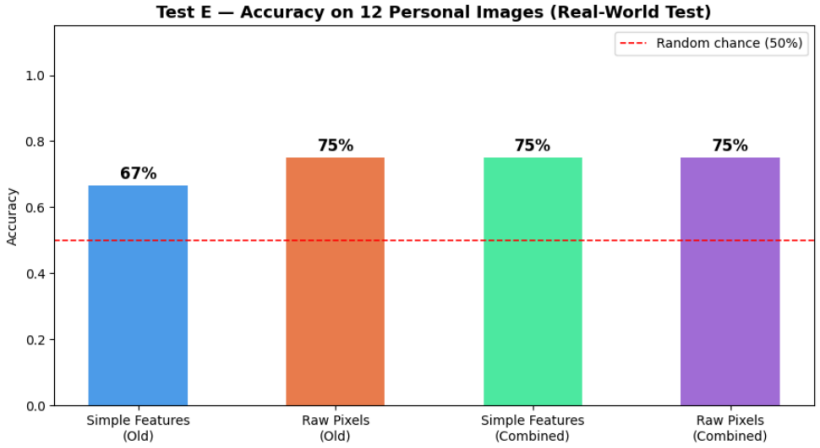

Figure 12: Accuracy of all four models on the 12-image personal test set.

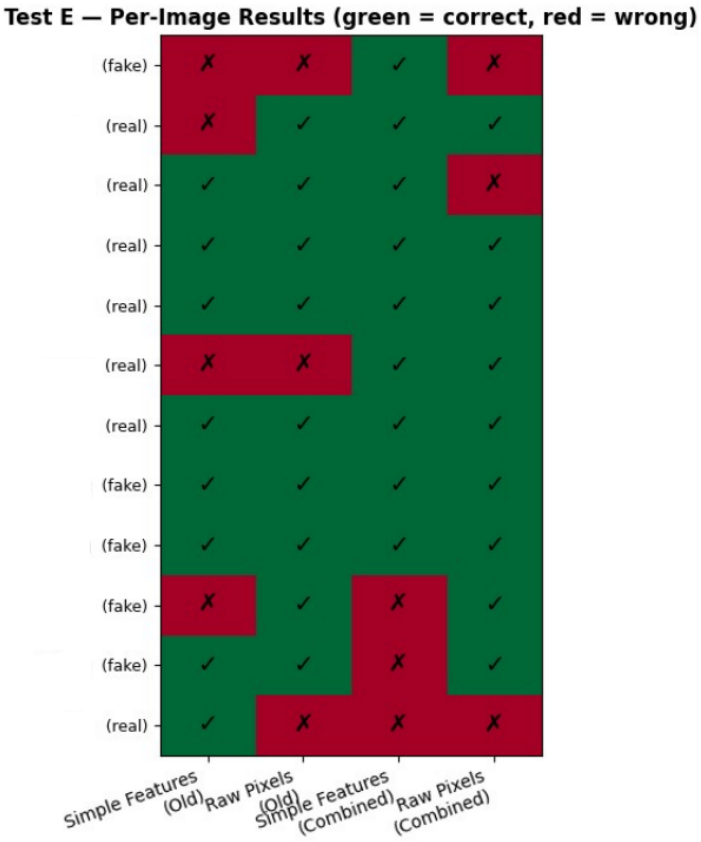

Figure 13: Per-image results across all four models. Green cells indicate a correct prediction, red cells an incorrect one.

4.4.4 Test F (Real Social Media Compression)

Finally, to test something closer to how images actually travel online, we used six image pairs, each consisting of the same photo before and after genuine social media compression, and evaluated all four models on both the original and the compressed version of each image.

Table 6: our 6 image pairs, original vs. compressed (Test F)

Model	**Original**	**Compressed**
Simple Features (Old)	50.0% (3/6)	33.3% (2/6)
Raw Pixels (Old)	50.0% (3/6)	50.0% (3/6)
Simple Features (Combined)	33.3% (2/6)	33.3% (2/6)
Raw Pixels (Combined)	50.0% (3/6)	50.0% (3/6)

Across this small set, compression alone did not change most predictions. Only one image, out of six, saw a model's prediction flip after compression: a real photo that the old Simple Features model correctly labelled as real in its original form, but mislabelled as fake after

compression. All other predictions, across all four models, stayed the same before and after compression.

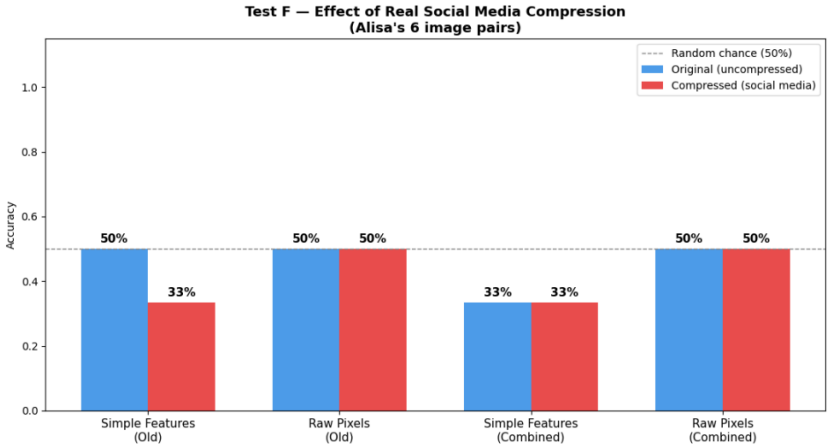

Figure 14: Accuracy on original vs. compressed images, across all four models.

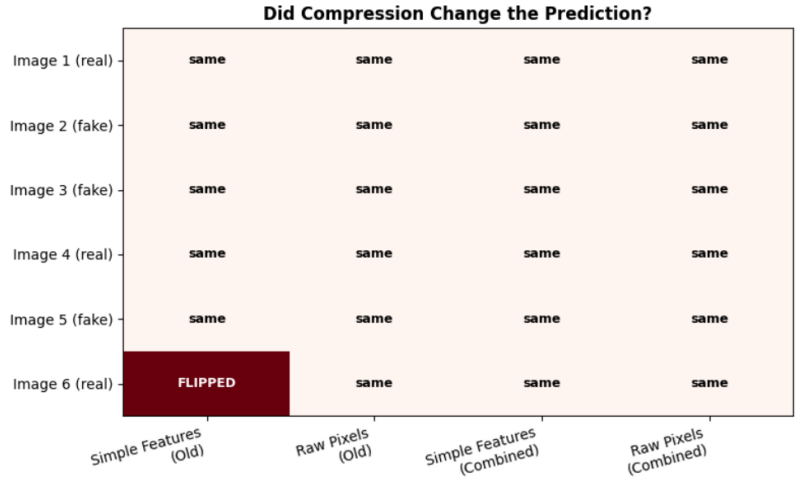

Figure 15: Whether each model's prediction changed after compression, for each of the six image pairs. Only one prediction, out of twenty-four, flipped.

4.5 What This Round of Experiments Suggests

Taken all together, these experiments point toward a consistent, if unglamorous, conclusion: accuracy on the dataset a model was trained on says very little about how that model will perform on images from outside that dataset. Both the Random Forest models and, independently, EfficienttNet model, showed strong performance on their respective evaluation sets and a meaningful drop when tested on more realistic, external images. In both cases, the leading explanation shifted, over the course of the project, from "the model overfit" toward "the dataset, both for training and for evaluation, was not representative enough of what real-world images actually look like". We are not claiming

this explains the full picture. But it was the most consistent pattern to emerge across every experiment in this study, and it is why the second round of experiments focused on dataset composition rather than on making either model more complex.

5 What We Learned: Discussion

When looking back at every experiment we ran, one thread runs through all of them, which is, a model's performance on the data it was trained on tells us very little about how it will behave in the real world. This was true for a simple Random Forest working with six numbers, it was true for the same Random Forest working with raw pixels, and it was true, independently, for a deep convolutional network fine-tuned on tens of thousands of images. Three very different levels of complexity, and the same failure mode kept appearing.

It looked at first like a it was about overfitting, about models memorising specific artifacts from specific AI generators rather than learning a general notion of "AI-generated". That explanation is not wrong, but over the course of this project it became clear that it is incomplete. When we looked more closely, both in our simplest models as well and in Neural Networks, the deeper issue was not just what the models learned, but what they were asked to learn from in the first place. The Kaggle dataset's real images were polished, professionally shot, closer to stock photography than to what a phone camera actually produces. Its fake images came from a narrow set of generators, some of them already outdated by the time we tested against them. A model trained on this combination does not need to learn "real versus fake" in any general sense. It only needs to learn "this particular kind of polish versus this particular kind of glitch", and that is a much easier, much less useful thing to learn.

This reframing changed what "improving the model" meant for the rest of the project. This time instead of reaching for a more complex architecture or more training epochs, we reached for more realistic data, naturalistic real photographs through the Shutterstock dataset and COCO, and more modern, harder AI-generated images that no longer carry the obvious tells of older generators. The results were not dramatic, Test C's Raw Pixels model reached 71.5% on the combined dataset, still far from reliable, but they were consistent. More diverse training data helped more than anything else we tried. It is an honest conclusion, and it is the conclusion the data actually supports.

A second thread worth mentioning is how differently Random Forest and EfficientNet fail. The Random Forest's mistakes are, in principle, inspectable, we can look at which feature it split on and at what threshold, and reason about why a particular image was misclassified. EfficientNet's 93% F1 score, by contrast, told us almost nothing about *why* it worked as well as it did on the Kaggle set, or why it failed on a personal photograph

with only 65.67% confidence. This is not a criticism of either method but a reminder that a model's accuracy and a model's transparency are separate questions, and a project like this one has to care about both, not just the first one.

Finally, this project reinforced something that has nothing to do with code. Two teams working somewhat independently, one with decision trees and handcrafted features, one with a pretrained neural network, arrived at the same diagnosis through different routes, which was that the dataset matters more than the model. That convergence, reached separately rather than assumed from the start, is probably the single most useful thing this study produced.

None of this amounts to a solution to AI image detection. It amounts to a clearer picture of why the problem is harder than it first appears, and where the next serious attempt at it should actually focus its effort.

6 Conclusion

This project set out to understand, from the ground up, what it means for an image to be AI-generated and what it takes to detect that. We did not set out to build the best possible detector. We set out to understand the problem honestly, to test our assumptions against real data, and to be transparent when those assumptions turned out to be wrong. Across a Random Forest built on six intuitive numbers, a Random Forest built on raw pixels, and a fine-tuned EfficientNet-V2-S model, we found the same pattern again and again, strong performance on a familiar dataset, and a meaningful drop when the images came from somewhere else. Understanding why that drop happens, and what it says about the datasets we build these models on, became the real subject of this paper.

What we take away from this project extends beyond the numbers in the previous section. The AI & Society Micro-Degree at the University of Graz did not just give us the technical tools to build and evaluate these models. It gave us the habit of asking why a result looks the way it does before accepting it, of testing an assumption instead of trusting it, and of treating a surprising or disappointing result as something worth explaining rather than something to hide. That habit, more than any specific technique, is what we consider the actual outcome of this micro-degree. At the same time, the program pushed us to think about this technology beyond its mechanics, about who is affected when detection fails, the legal and ethical frameworks still catching up to what these models can already do, and what it means for a society when the line between real and generated images becomes harder to see. We leave this program not only more technically capable, but more aware of the responsibility that comes with that capability.

We would like to thank the University of Graz and the AI & Society Micro-Degree for

the opportunity to carry out this project, and the professors of the program for the breadth and depth of what they taught us, both technical and societal, over the course of this degree. We are especially grateful to Professor Di Natale Anna, who supervised this project. Her guidance helped this work at every stage, and her encouragement, even when our results were messy or our conclusions uncertain, kept us optimistic and motivated to keep exploring rather than settle for an easy answer.

Code and Data Availability

Codes for this project are available at https://github.com/SamahG13/ai-image-detection and https://github.com/DNeberize/ai_vs_real. The repositories include the notebooks used to train and evaluate the Random Forest and EfficientNet-V2-S models, along with a note on how the code was developed. Datasets used include the *Kaggle AI-Generated vs. Real Images* dataset and the *Shutterstock AI vs. Human-Generated Images* dataset, both publicly available on Kaggle.

References

[1] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[3] European Parliament and Council of the European Union. Regulation (eu) 2024/1689 of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act). https://eur-lex.europa.eu/eli/reg/2024/1689/oj, 2024. Official Journal of the European Union.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[5] Markus Kneer, Michele Loi, and Markus Christen. Trust and responsibility in human-ai interaction, 2024. University of Zurich / IDea Lab, University of Graz.

[6] Jana Lasser, Segun T. Aroyehun, Fabio Carrella, Almog Simchon, David Garcia, and Stephan Lewandowsky. From alternative conceptions of honesty to alternative facts in communications by u.s. politicians. *Nature Human Behaviour*, 2023.

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[8] Martin Miernicki and Irene (Huang Ying) Ng. Artificial intelligence and moral rights. *AI & Society*, 36(1):319–329, 2021.

[9] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[10] Mingxing Tan and Quoc V. Le. Efficientnetv2: Smaller models and faster training. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.