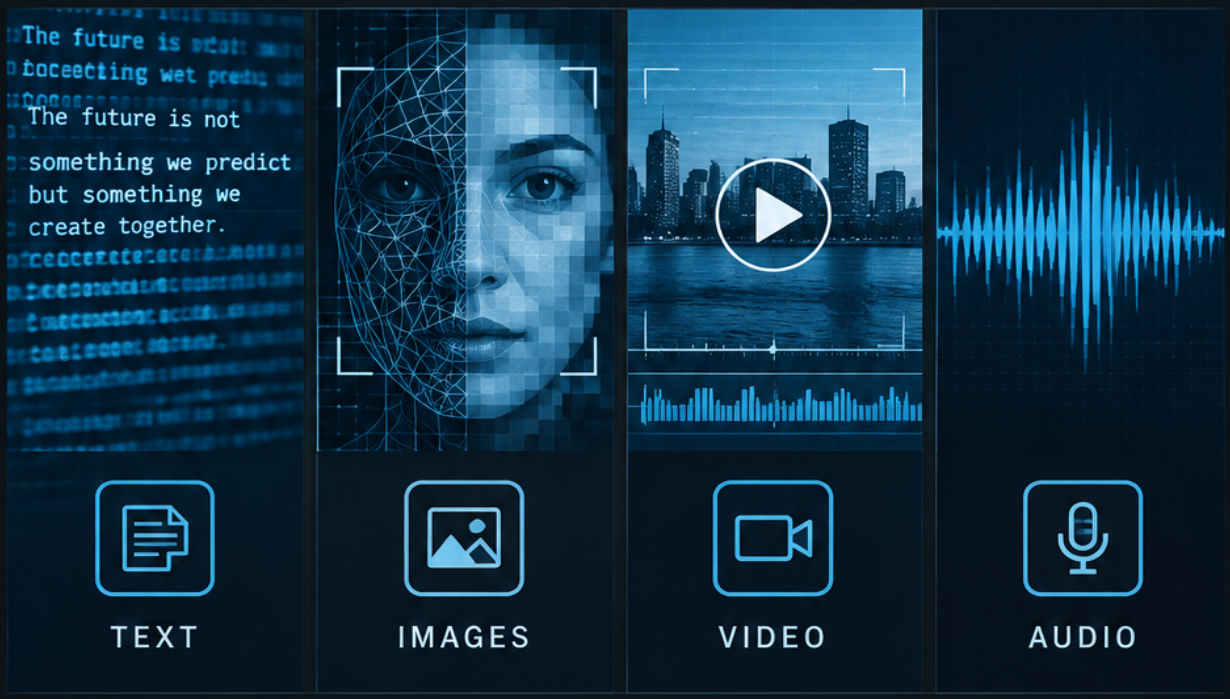
THE

AI FORENSICS HANDBOOK

Detection and Analysis of Artificially Generated Text, Images, Video, and Audio



JEFFREY C. MORGAN

The AI Forensics Handbook

Detection and Analysis of Artificially Generated Text, Images, Video, and Audio

Steve Publications

This book is available at https://leanpub.com/theaiforensicshandbook

This version was published on 2026-09-10

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2026 Steve Publications

Contents

Detection and Analysis of Artificially Generated Text, Images, Video, and Audio

This book provides a complete, technically rigorous treatment of how AI-generated content can be detected, analyzed, and attributed. It explains the generative models that produce synthetic media, the forensic signals they leave behind, and the mathematical, algorithmic, and empirical methods used to identify them. Each chapter combines foundational theory with practical implementation, and every approach is examined not only for what it detects but for why it works, when it fails, and under what assumptions it remains valid. The reader will finish with both the scientific understanding to evaluate any detection claim and the practical ability to build, test, and deploy detection systems.

Introduction

The ability of artificial intelligence to generate text that reads naturally, images that look real, voices that sound authentic, and videos that show convincing faces has moved from laboratory demonstrations to everyday use within a few years. In 2022 alone, the proliferation of publicly available large language models and diffusion-based image generators made high-quality synthetic content accessible to anyone with an internet connection. By 2024, systems that generate coherent video sequences and photorealistic talking heads had emerged, closing what many thought would be a durable gap between artificial and human-created media.

This capability is not inherently harmful. The same models that can forge political speeches or financial reports can help writers draft emails, students explore ideas, artists iterate designs, and developers prototype user interfaces. Yet the capacity for misuse is real and growing. A 2024 study of in-the-wild deepfakes found synthetic content circulated across 88 websites in 52 languages, with hours of manipulated video and audio disseminated daily on social platforms. Fraud schemes using cloned executive voices have already cost organizations tens of millions of dollars. Political campaigns face the prospect of convincing fake speeches. Journalists and historians confront the possibility that the public record can be contaminated with fabricated images and documents.

Detection is the obvious first response. If we can tell whether content is AI-generated, we can warn users, filter suspicious material, attribute responsibility, and preserve trust in information ecosystems. But detection is not straightforward, and much of the public discussion about it is dangerously oversimplified. Headlines proclaim that detectors can spot AI text with high accuracy, while research papers simultaneously demonstrate that those same detectors fail catastrophically when evaluated on different models, different domains, or simple transformations like paraphrasing. The gap between benchmark performance and real-world performance is large and persistent.

This book is built on a central thesis: AI-content detection is a statistical and forensic problem, not a product category. There is no detector that works universally, no single signal that distinguishes human from machine content

across all modalities, and no method that is immune to change as generative systems improve. Every detection technique exploits specific signals tied to particular generative mechanisms under particular assumptions. Understanding those signals, mechanisms, and assumptions is what separates responsible practice from cargo cult science.

Consider text. A detector might look for low perplexity: the statistical property that each word is highly predictable given the words before it. Large language models select tokens to maximize probability under their learned distributions, so the text they produce tends to be more predictable than human writing, which contains idiosyncrasies, digressions, and unexpected turns. This signal is real and measurable. But it is not universal. A non-native English writer often produces low-perplexity text. A technical manual writer produces low-perplexity text. A student carefully following a template produces low-perplexity text. And a sufficiently advanced model trained on diverse human writing can produce high-perplexity text that matches human variability. The perplexity signal is not wrong, but it is limited, and any detector that treats it as definitive will make systematic errors.

Consider images. Generative adversarial networks leave frequency-domain artifacts caused by their upsampling operations. These artifacts are consistent across different GAN architectures and datasets, and classifiers trained on frequency representations can detect GAN-generated images with remarkable accuracy. But diffusion models, which dominate modern image generation, leave different artifacts: reduced high-frequency content, latent-space fingerprints, and reconstruction behaviors that reflect their iterative denoising process. A detector trained on GAN artifacts will miss diffusion-generated images. A detector trained on diffusion artifacts may fail on future models that use different architectures or training procedures. The signal changes with the technology.

Consider video. Deepfake videos often exhibit temporal inconsistencies: flickering at face boundaries, unnatural eye movements, mismatched lip-sync. Frame-based detectors look for these cues and achieve high accuracy on datasets like FaceForensics++. But when evaluated on in-the-wild deepfakes collected from the internet in 2024, the same detectors showed AUC scores that dropped by approximately 50 percent compared to benchmark performance. The gap reflects compression, resolution changes, lighting conditions, and improved generation methods that all degrade or mask the signals the detector learned to find.

Consider audio. Neural vocoders, which convert spectrograms into waveforms in modern text-to-speech and voice-cloning systems, leave subtle artifacts in the audio spectrum. Detection systems can learn to identify these artifacts and distinguish synthetic speech from human speech with high accuracy on curated datasets. But when the same systems are evaluated on data recorded through different microphones, transmitted over phone lines, or compressed by messaging apps, their performance degrades significantly. The artifacts that are clear in a controlled setting are drowned out by the noise and distortion of real-world conditions.

These examples illustrate three fundamental points that recur throughout this book.

First, detection is signal-driven. Every detector works by measuring properties of the input and comparing them against expected distributions for human-created versus machine-generated content. Those properties might be statistical (perplexity, entropy, frequency spectra), structural (attention patterns, compression artifacts, biometric cues), semantic (coherence, factual accuracy, cross-modal consistency), or cryptographic (watermarks, digital signatures). The detector's job is to measure, compare, and decide. Understanding what property is being measured and why it differs between human and machine content is essential to understanding what the detector can and cannot do.

Second, detection is assumption-dependent. Every signal relies on assumptions about the generative process, the distribution of the data, and the conditions under which the content was captured, transmitted, and stored. If those assumptions are violated, the signal may disappear or become misleading. A perplexity-based detector assumes that human writing and model-generated writing have different predictability profiles. A frequency-domain image detector assumes that the image has not been heavily compressed or filtered. A deepfake video detector assumes that the manipulation leaves temporal artifacts that are visible at the given resolution. When assumptions fail, detectors fail.

Third, detection is probabilistic. No detector outputs a ground-truth label. It outputs a score, a probability, or a confidence that can be calibrated, thresholded, and interpreted, but never guarantees correctness. This probabilistic nature is not a bug, it is a consequence of the underlying mathematics. If human and machine distributions overlap significantly (which they increasingly do), then any decision rule will have both false positives and false negatives.

The task of detection is not to eliminate these errors but to understand their distribution, minimize them where they are costly, and communicate their existence clearly to whoever relies on the detector's output.

The chapters that follow develop these ideas systematically. We begin with the foundations: how generative models work, how they learn to produce content, and where their artifacts originate. We then move through each modality in turn, building from classical forensic methods to modern deep learning approaches, explaining the architecture of representative detectors and providing functional code implementations. We treat evaluation as a major subject, showing how to measure performance honestly and avoid the traps of benchmark leakage, dataset contamination, and overfitting. We examine adversarial dynamics, explaining how detectors can be evaded and how to design systems that are more robust. Finally, we place detection within real-world forensic workflows, where probabilistic evidence must be combined with human judgment, procedural rigor, and clear communication about what can and cannot be concluded.

Throughout, we maintain a critical stance. When evidence is strong and replicated, we say so. When findings are preliminary or disputed, we say so. When a technique is promising but not yet reliable, we say so. When a claim in the literature appears to be overstated or based on flawed methodology, we flag it. The goal is not to be pessimistic about detection but to be honest about what it can achieve, because the stakes are too high for anything less.

Chapter 1: Generative AI Foundations

Every detection technique rests on an understanding of what it is detecting. To know what signals a generative model leaves in its output, we must first understand how the model produces that output in the first place. This chapter establishes the probabilistic, optimization-theoretic, and architectural foundations that underlie all modern generative AI systems. These foundations are not specific to any single modality; they apply to text, images, audio, video, and multimodal generation alike. Grasping them will make every subsequent chapter easier to understand, because the artifacts we look for are direct consequences of the mechanisms described here.

Probability Distributions and Sampling

At its core, a generative model is a probability distribution over possible outputs. When we say "this model generates images," we mean that it defines a probability distribution p(x) over the space of all possible images x, and that sampling from this distribution produces images that look like the training data.

Consider a simple discrete example. Suppose we want to model the distribution of English letters in a text corpus. We count the occurrences of each letter and normalize to get probabilities: p(a) = 0.082, p(b) = 0.015, and so on. To generate text from this distribution, we sample letters one by one. The result is a random string whose letter frequencies match English, but which is not meaningful because we are ignoring context.

Generative models extend this idea by modeling conditional distributions. Instead of p(letter), we model p(letter | previous letters). A sufficiently powerful model that captures the full conditional distribution of letters given their context could, in principle, generate text that is indistinguishable from human writing. The challenge is that the space of possible outputs grows exponentially with sequence length, and the conditional distributions are far more complex than letter frequencies.

For continuous outputs like images or audio, the concept is the same but the mathematics uses continuous distributions. An image might be represented as

a vector of pixel intensities x in R^(H×W×C), where H, W, and C are height, width, and number of color channels. A generative model defines p(x) over this continuous space. Sampling produces a pixel vector that, when reshaped, looks like a plausible image.

The distribution p(x) is not given; it is learned from data. We observe a dataset D = {x_1, x_2, ..., x_N} of real examples (texts, images, audio clips) and assume they are drawn from some unknown true distribution p_data(x). The generative model parameterizes a family of distributions p_theta(x), where theta represents the model's learnable parameters. Training adjusts theta so that p_theta(x) approximates p_data(x) as closely as possible.

How close is close enough? The answer depends on how the model is evaluated. If we measure similarity using the Kullback-Leibler divergence:

```
KL(p_data || p_theta) = sum_x p_data(x) log(p_data(x)/p_theta(x))
```

then minimizing KL divergence penalizes the model heavily for assigning low probability to data points that are actually common. This is called the "zero-forcing" behavior of KL divergence, because the model tries to cover every mode of the true distribution to avoid regions where p_data(x) > 0 but p_theta(x) ≈ 0.

If we instead minimize the reverse KL divergence:

```
KL(p_theta || p_data) = sum_x p_theta(x) log(p_theta(x)/p_data(x))
```

the model exhibits "mode-seeking" behavior. It concentrates probability mass on the most common regions of p_data and may ignore rare modes entirely. This is a characteristic of many GAN-based methods, which we discuss in Chapter 3.

Understanding these divergence properties matters for detection. A mode-seeking model might generate only stereotypical images and miss edge cases, while a mode-covering model might generate plausible-but-unusual content. Either behavior can create detectable patterns, depending on what the detector is looking for.

Likelihood, Loss Functions, and Optimization

Generative models learn by maximizing the likelihood of the training data under the model's distribution. Maximum likelihood estimation asks: what choice of theta makes the observed data most probable? Mathematically, we maximize:

```
L(theta) = product_{i=1}^N p_theta(x_i)
```

or equivalently, maximize the log-likelihood:

```
log L(theta) = sum_{i=1}^N log p_theta(x_i)
```

Different generative families define p_theta(x) in different ways, leading to different practical training objectives.

For autoregressive models (including modern language models and some audio models), the distribution is factorized as:

```
p_theta(x) = product_{t=1}^T p_theta(x_t | x_1, ..., x_{t-1})
```

where x_1, ..., x_T is the sequence of tokens or samples. Training optimizes the negative log-likelihood of each token given its predecessors, which reduces to a cross-entropy loss at each position. This is exactly the objective used to train GPT-family models.

For diffusion models, training minimizes a variational lower bound on the negative log-likelihood. The objective can be approximated as a simple noise-prediction loss at each timestep, which is why diffusion training is surprisingly straightforward despite the complex generative process.

For GANs, there is no explicit likelihood. Instead, the model is trained by a min-max game between a generator that tries to produce samples indistinguishable from real data and a discriminator that tries to tell real from fake. The training objective is:

```
min_G max_D E_{x~p_data}[log D(x)] + E_{z~p_z}[log(1 - D(G(z)))]
```

where G is the generator, D is the discriminator, and z is a random latent vector. This objective does not correspond to maximizing likelihood directly, which is part of why GAN training is notoriously unstable and why the generated samples do not admit easy probability calculations.

These different optimization objectives leave different signatures. Autoregressive models optimize token-by-token probability, which influences their statistical properties. Diffusion models optimize noise prediction, which influences their spatial frequency characteristics. GANs optimize adversarial confusion, which influences their structural coherence but leaves characteristic artifacts. Detection methods exploit these differences.

The Generation Pipeline from Model to Output

Understanding the complete pipeline from a trained model to its final output reveals additional sources of detectable variation. A trained model is only the first stage; how it is used to produce content introduces parameters and procedures that affect the output's forensic profile.

For an autoregressive language model, the generation pipeline consists of:

1. Encoding the input prompt into token embeddings.
2. Running the model's transformer layers to compute probability distributions over the vocabulary at each step.
3. Selecting the next token from the distribution using a decoding strategy.
4. Repeating steps 2-3 until a stopping condition is met.

The decoding strategy is critical. If the model always selects the highest-probability token (greedy decoding), the output will be highly predictable and lack variation. If the model samples randomly from the full distribution, the output may be more diverse but also more erratic. In practice, models use temperature-scaled sampling or constrained sampling (top-k, top-p) to balance coherence and variation.

Temperature scaling modifies the probability distribution by dividing log-probabilities by a parameter T before applying the softmax function. Lower temperature ($T < 1$) sharpens the distribution, making high-probability tokens

even more likely. Higher temperature ($T > 1$) flattens the distribution, making rare tokens more likely. Different temperatures produce outputs with different statistical properties, which detection methods can exploit.

Top-k sampling restricts sampling to the k most probable tokens, renormalizing their probabilities. Top-p (nucleus) sampling restricts sampling to the smallest set of tokens whose cumulative probability exceeds p. Both methods prune unlikely tokens and create characteristic patterns in the output that differ from both pure greedy decoding and pure random sampling.

For image generators based on diffusion models, the pipeline includes:

1. Starting with random Gaussian noise.
2. Iteratively denoising the image using the trained model for T steps (where T might be 20-1000 depending on the method).
3. Optionally conditioning each step on a text prompt or other input.
4. Optionally applying guidance methods like classifier-free guidance to steer the output.

The number of sampling steps, the choice of sampling algorithm (DDPM, DDIM, etc.), the guidance scale, and the random seed all affect the output. Different choices leave different artifacts. A diffusion image generated in 50 steps may have different characteristics than the same image generated in 1000 steps, because the shorter process leaves residual noise structure that would be removed in longer sampling.

For GAN-based image generators, the pipeline is simpler: sample a latent vector z and run it through the generator network G to produce G(z). The generator is a deterministic function, so the output is fully determined by the latent vector. This simplicity is both a strength (fast generation) and a weakness (the latent space structure and generator architecture leave clear fingerprints).

Every stage in these pipelines introduces parameters that affect the output's forensic profile. Detection systems that ignore these parameters and treat "AI-generated" as a monolithic category miss opportunities to exploit variation. For example, a detector trained only on outputs from greedy decoding may fail on outputs from high-temperature sampling, because the statistical patterns are different.

Where Artifacts Originate in Generative Models

Generative artifacts arise from fundamental constraints and design choices in the models themselves. They are not bugs but consequences of how the models work. Understanding these origins is essential to understanding which signals are robust and which are fragile.

Approximation error is a universal source of artifacts. No generative model perfectly captures the true data distribution, because the model has finite capacity and the true distribution is infinitely complex. The mismatch between model distribution and data distribution manifests as artifacts: images that look almost right but contain impossible geometries, text that is fluent but factually incoherent, audio that sounds natural but has subtle timing irregularities.

Architecture-imposed constraints create systematic artifacts. Convolutional neural networks operate with local receptive fields and weight sharing, which means they learn patterns that are translationally invariant but may miss long-range dependencies. Upsampling operations (transposed convolutions, nearest-neighbor upsampling, bilinear interpolation) introduce characteristic patterns. Transposed convolution with uneven kernel sizes and stride creates checkerboard artifacts: a regular grid-like pattern in the output that reflects how the upsampling layer tiles its receptive fields across the spatial dimensions.

The tokenization process in language models creates a discretization artifact. Text is split into subword units (tokens), each of which is assigned a probability. The model optimizes token-level likelihood, not character-level likelihood or semantic-level coherence. This means the model can produce text that is statistically fluent at the token level while containing semantic inconsistencies or awkward boundaries between tokens. Detection methods that operate at the character or syllable level can sometimes find signals that token-level models miss.

Latent-space structure is another source. Models that generate by transforming a latent vector (VAEs, GANs, diffusion models in latent space) operate in a compressed representation. The mapping from latent space to output space cannot be perfectly bijective, so different regions of latent space map to outputs with different characteristics. Some regions produce high-quality outputs; others produce artifacts or collapse. If a detector learns to recognize

the boundary between high-quality and low-quality latent regions, it can exploit this structure.

Iterative refinement processes leave temporal artifacts. Diffusion models generate images through hundreds of denoising steps. Each step removes a portion of the noise and adds structure guided by the learned model. If the process is stopped early, residual noise patterns remain. If the process uses a simplified sampling method, the trajectory through noise space may differ from the training distribution, creating subtle inconsistencies. These artifacts are not random; they reflect the specific algorithm used.

Model Capacity, Overfitting, and Memorization

A generative model's capacity (the amount of information it can store and represent) directly affects the nature of the artifacts it produces. Small models have insufficient capacity to capture the complexity of the data distribution, so they produce oversimplified, stereotypical, or blurry outputs. Large models can capture fine detail and diversity, but they risk overfitting to the training data.

Overfitting in generative models manifests as memorization: the model produces outputs that are essentially copies or near-copies of training examples rather than genuine generalizations. This is particularly concerning for language models trained on public text, which can reproduce copyrighted material, personal information, or code snippets verbatim.

Memorization is detectable. A model that is memorizing rather than generalizing will produce outputs that:

- Have unusually high likelihood under the model (because they match training data exactly)
- Appear in external databases (because the training data is often publicly available)
- Lack the variability characteristic of true generation (repeated prompts yield nearly identical outputs)

Detection systems can use retrieval-based methods: given a candidate text or image, search for similar content in known corpora. A match suggests memorization. This is not a universal detection method, because models can

generalize well while still producing coincidentally similar outputs. But it is a useful tool for identifying certain kinds of synthetic content.

The relationship between capacity, overfitting, and detection is nuanced. A detector trained on outputs from a small, underfitting model may look for artifacts like blurriness or stereotypical patterns. Those artifacts may not appear in outputs from a larger, better-fitting model, so the detector fails to generalize. Conversely, a detector trained on a model that memorizes heavily may learn to identify memorization-specific patterns that do not appear in well-regularized models. This is one reason why cross-model generalization is so difficult: different models operate at different points in the capacity-overfitting trade-off space, and each point has different artifacts.

Transfer Learning and Its Forensic Implications

Most modern generative models use transfer learning: they are pretrained on large, general datasets and then fine-tuned or adapted for specific tasks or styles. This practice has important forensic implications.

Pretraining establishes a base distribution. A language model pretrained on Wikipedia, Common Crawl, and books learns a general distribution over language that reflects the statistical properties of those sources. Fine-tuning on a specific domain (medical text, code, creative writing) shifts the distribution but does not erase the pretraining. The resulting model's outputs contain signals from both stages.

This layering creates a forensic signature. A detector that understands the characteristics of the base model (its tokenization, its vocabulary, its tendency toward certain syntactic patterns) can sometimes identify content as coming from that family of models even after fine-tuning. For example, all models in the GPT family share a similar tokenization scheme and architectural structure, so outputs from different GPT variants share low-level statistical properties that may be detectable.

Fine-tuning changes the distribution in directionally predictable ways. Instruction-tuning makes models more compliant and structured. Role-playing fine-tuning makes them more stylized. Domain adaptation makes them more specialized. Each change modifies the statistical profile of the output. A detector trained to recognize the base distribution may not recognize the fine-tuned distribution, and vice versa.

This layering is particularly relevant for attribution. If we can distinguish between outputs from a base model and outputs from its fine-tuned variants, we can make progress on source identification: determining not just that content is AI-generated, but which model generated it. This is challenging because fine-tuning can erase distinguishing features, and multiple fine-tuned models may share the same base. But it is not impossible, and we return to it in Chapter 16.

The broader lesson from transfer learning is that generative models are not atomic. They are composites, built from stages of training and adaptation that leave cumulative signatures. Forensic analysis that treats them as single units misses the richness of these layered signatures.

Chapter 2: Language Models and Text Generation

Text generation is the modality where AI capabilities advanced most dramatically and where detection became most urgently needed. The release of increasingly capable large language models in 2022 and 2023 made high-quality AI text freely available to millions of users, and the resulting flood of synthetic content in academic, professional, and public contexts made detection a practical necessity. Yet text is also the modality where detection is most unreliable, because language models are so good at mimicking human statistical patterns. This chapter explains why: we examine the internal architecture of transformers, the mechanics of text generation, and the specific signals that distinguish model-generated text from human writing.

Transformer Architecture: Attention, Layers, and Representations

Modern language models are built on the transformer architecture, introduced by Vaswani et al. in 2017 in the paper "Attention Is All You Need." The transformer was originally designed for machine translation but became the foundation for all large language models, from BERT and GPT to LLaMA and beyond. Its key innovation was the self-attention mechanism, which allows every position in a sequence to directly attend to every other position, regardless of distance.

Here is how self-attention works, in technical detail. Given an input sequence of token embeddings X in R^(n×d), where n is the sequence length and d is the embedding dimension, the attention mechanism computes three projections:

```
Q = XW_Q
K = XW_K
V = XW_V
```

where W_Q, W_K, and W_V are learned weight matrices in R^(d×d_k), R(d×d_k), and R^(d×d_v) respectively. These are the query, key, and value matrices. The attention output is:

```
Attention(X) = softmax(QK^T / sqrt(d_k)) V
```

The softmax operates on the dot products between queries and keys, producing attention weights that sum to one for each query position. These weights determine how much each value contributes to each position's output. Intuitively, each token generates a query asking "what other tokens are relevant to me?" and compares it against all keys to find matches.

Multi-head attention runs this process multiple times in parallel with different weight matrices, producing different attention "perspectives." The outputs of all heads are concatenated and projected:

```
MultiHead(X) = Concat(head_1, ..., head_h)W_O
```

where each head_i = Attention(XW_Q^i, XW_K^i, XW_V^i).

A transformer decoder layer (used in models like GPT) consists of:

1. A masked multi-head self-attention block (masked so each position can only attend to previous positions)
2. A feed-forward network applied independently at each position
3. Layer normalization and residual connections around each block

Stacking many layers (modern models use 30-100+) creates a deep network that transforms the input embeddings through many rounds of attention and nonlinear computation. The final layer's output at position t is a vector that encodes the model's understanding of the sequence up to that point, which is then projected to a vocabulary distribution for next-token prediction.

This architecture has forensic implications. The attention mechanism allows the model to capture long-range dependencies and complex patterns,

which makes its outputs more human-like than outputs from recurrent models. But it also means the model learns statistical regularities in attention patterns: certain token relationships become reinforced, certain syntactic structures become preferred, and certain semantic associations become strong. These learned patterns are not identical to human language processing, and they can create detectable signatures.

For example, transformer models tend to produce text with very specific attention-driven coherence: the text is locally consistent (each token attends appropriately to its context) but may lack global coherence (the overall narrative or argument may wander, contradict, or repeat). This is because the model optimizes local probability, not global structure. Detection methods that measure global coherence or narrative consistency can sometimes exploit this gap.

Another consequence is that the model's behavior is determined by its parameter values, which are fixed after training. This means the model's statistical patterns are stable and reproducible. Two different prompts may produce text with very different surface content, but the underlying statistical properties (token probability distributions, syntactic patterns, stylistic tendencies) will be consistent. A detector trained on enough examples can learn these stable patterns.

Tokenization, Vocabulary, and Decoding Strategies

Before a language model can process text, the text must be tokenized: split into subword units that the model recognizes. Different models use different tokenizers, and the tokenizer is not a neutral preprocessing step; it shapes the model's behavior and leaves detectable signatures.

The most common tokenization methods are Byte-Pair Encoding (BPE) and its variants. BPE starts with a base vocabulary of characters and iteratively merges the most frequent adjacent pairs until the vocabulary reaches a target size. For example, the string "unbelievable" might be tokenized as ["un", "believ", "able"] or ["un", "believ", "ab", "le"] depending on the training data and vocabulary size.

The GPT-family models use BPE-based tokenizers with vocabulary sizes ranging from 50,000 to 100,000 tokens. LLaMA uses a similar approach. These tokenizers are trained on the model's training corpus, so the vocabulary

reflects the statistical patterns of that corpus. Rare words, non-English characters, and novel combinations may be split into many tokens, while common words may be single tokens.

Tokenization affects detection in several ways:

- Token boundaries create a discretization that can be visible in the output. A model that optimizes at the token level may produce text where the token boundaries are statistically coherent but the character-level or syllable-level patterns are not.
- Different models use different tokenizers, so outputs from different models have different tokenization signatures. A detector that operates at the token level (using token probabilities, for example) is implicitly tied to a specific tokenizer.
- The model's vocabulary limits what it can express directly. If a word is not in the vocabulary, the model must express it as a sequence of subword tokens, which may affect fluency and statistical patterns.

After tokenization and model processing, the next step is decoding: selecting the actual output tokens from the model's probability distributions. As noted in Chapter 1, the decoding strategy profoundly affects the output.

Greedy decoding always selects the highest-probability token. The result is text that is coherent but often repetitive and unnatural, because the model gets stuck in local probability maxima.

Temperature-scaled sampling modifies the logits (log-probabilities) by dividing by a temperature T before applying softmax. The modified probability is:

```
p'(x_t) = exp(z_t/T) / sum_j exp(z_j/T)
```

where z_t is the logit for token t. With T = 0.7, for example, the distribution is sharpened compared to T = 1.0, making the output more conservative and predictable. With T = 1.5, the distribution is flattened, making the output more creative and variable.

The forensic consequence is clear: outputs with different temperatures have different statistical profiles. Low-temperature outputs have lower perplexity and less variation in sentence structure. High-temperature outputs

have higher perplexity and more variation. A detector calibrated for one temperature may misclassify outputs from another.

Top-k sampling selects from only the k most probable tokens. This creates a hard cutoff that affects the tail of the distribution: rare tokens are never selected, even if they would be stylistically appropriate. The result is text that avoids unusual word choices, which may be detectable as an absence of certain token types.

Top-p sampling (nucleus sampling) selects from the smallest set of tokens whose cumulative probability exceeds p. With p = 0.9, for example, the model samples from the top 90 percent of probability mass. This adapts to the local distribution: in uncertain positions, more tokens are included; in confident positions, fewer. The effect is more natural variation than top-k, but it still prunes the tail.

Different models use different default decoding parameters. OpenAI's GPT models typically use temperature around 0.7 and top-p around 0.9 for chat. Other providers use different defaults. Users can override defaults, creating further variation. A detection system must account for this parameter space.

Next-Token Prediction and Probability Distributions

At the heart of autoregressive language models is next-token prediction: given a sequence of tokens, predict the probability distribution over the next token. This is a seemingly simple task, but it is powerful enough to generate text that is coherent, contextually appropriate, and stylistically varied.

The model computes a distribution p(x_t | x_1, ..., x_{t-1}) at each step. This distribution reflects everything the model has learned about language from its training data. If the model is well-trained, the distribution assigns high probability to tokens that would naturally follow the prefix.

Detection methods exploit these probability distributions directly. The fundamental observation is that text generated by a model tends to have higher probability under that model's distribution than human text of similar content. This is because the model selects tokens to maximize probability (or sample from high-probability regions), while humans select tokens based on intentions, knowledge, and stylistic preferences that may not align with statistical optimality.

This observation leads to likelihood-based detection. Given a candidate text and a reference model, compute the log-likelihood of the text under the model:

```
log p_theta(x) = sum_{t=1}^T log p_theta(x_t | x_1, ..., x_{t-1})
```

If the log-likelihood is unusually high, the text is more likely to be model-generated. If it is low, the text is more likely to be human.

Likelihood-based detection has strengths and weaknesses. It is conceptually simple and requires no additional training. But it is also easily confounded: a human who writes in a statistical style will have high likelihood, while a model that generates diverse text will have lower likelihood. It is also computationally expensive for long texts, because it requires running the full model for each token.

Perplexity is a related measure. Perplexity is defined as:

```
PPL(x) = exp(-1/T * log p_theta(x))
```

It can be interpreted as the effective branching factor: the average number of equally likely choices the model considers at each step. Lower perplexity means the model is more confident in its predictions; higher perplexity means more uncertainty.

AI-generated text tends to have lower perplexity under the generating model than human text, because the model selects tokens near the peak of its probability distribution. But perplexity alone is not sufficient: many factors affect perplexity, including the text's topic, the model's familiarity with that topic, and the reference model used for computation. A perplexity-based detector must be calibrated carefully and used alongside other signals.

Temperature, Top-k, Top-p, and Their Forensic Consequences

The decoding parameters we discussed above do not just affect how text looks; they affect the forensic signals that detection methods measure. This is a critical point that is often overlooked in detection literature.

When a model uses low temperature, the output is concentrated near the mode of the probability distribution. This means:

- Perplexity is low (the text is highly predictable)
- Burstiness (variation in sentence length and complexity) is low
- Rare tokens are rarely used
- Repetition may increase

All of these are signals that detectors exploit. Low perplexity, low burstiness, absence of rare tokens, and repetition are hallmarks of AI-generated text, at least for models using default or low-temperature decoding.

When a model uses high temperature, the opposite is true:

- Perplexity is higher
- Burstiness is higher
- Rare tokens appear more often
- The text is more variable

A detector trained on low-temperature outputs may fail on high-temperature outputs, because the signals it learned (low perplexity, low burstiness) are absent. This is not a failure of the detector's architecture; it is a consequence of the detector being trained on a specific region of the parameter space.

This also means that users who want to evade detection can adjust decoding parameters to change the forensic profile of the output. Increasing temperature and using top-p sampling can make AI text statistically more similar to human text. This is not foolproof: advanced detectors may find other signals. But it is effective against simple detectors that rely primarily on perplexity and burstiness.

Similarly, different models have different inherent statistical profiles. Some models are trained with objectives that encourage diversity (penalizing repetition, encouraging exploration), while others are trained for coherence and faithfulness. A model that generates diverse text may be harder to detect using perplexity-based methods, because its outputs have higher perplexity.

Emergent Properties and Capability Scaling

As language models grow larger, they exhibit emergent properties: capabilities that are not present in smaller models and that cannot be predicted from the model's architecture and training objective alone. Emergent properties include:

- Instruction following: the ability to understand and execute complex instructions
- Chain-of-thought reasoning: the ability to solve problems by generating intermediate steps
- Multilingual competence: the ability to operate fluently in many languages
- Code generation: the ability to write correct, functional code
- Style transfer: the ability to mimic specific writing styles

These emergent properties make the models more useful, but they also make them harder to detect. A model that can follow instructions to "write in a casual, slightly imperfect style" will produce text that is deliberately different from its default style. A model that can reason through problems will produce text with logical structure that mimics human reasoning.

The scaling laws discovered by Kaplan et al. and follow-up work show that model performance improves predictably as a function of model size, dataset size, and compute. This means that as models grow, their statistical properties change in ways that are not obvious. A detector trained on outputs from a 7-billion-parameter model may not work on outputs from a 70-billion-parameter model, because the larger model's text has different statistical properties.

This has practical implications for detection system maintenance. Detection models must be continuously updated as new generation models appear. A one-time training is insufficient. And cross-model generalization, while desirable, requires methods that capture stable properties of generation rather than transient patterns of specific model versions.

Fine-Tuning, Prompting, and Behavioral Variation

A pretrained language model's default behavior can be modified through fine-tuning and prompting, and these modifications have forensic consequences.

Fine-tuning adjusts the model's weights on a specific dataset, changing its distribution over outputs. Instruction tuning (like RLHF: reinforcement learning from human feedback) trains the model to follow instructions and produce outputs that human raters prefer. The result is a model that is more helpful, more structured, and more aligned with human expectations.

The forensic consequence is that fine-tuning changes the statistical profile of the output. A model fine-tuned for helpfulness may produce more coherent, more grammatically correct text than the base model. It may also be more likely to include certain phrases ("Of course," "Here is the answer," "Let me help you") that become stylistic markers. A detector trained on base-model outputs may not recognize fine-tuned outputs, and a detector trained on fine-tuned outputs may misclassify base-model outputs.

Prompting is a non-parametric form of control. By changing the input prompt, we can change the model's output without changing its weights. "Write a formal essay" produces different text than "Write a casual blog post," even from the same model with the same weights.

The forensic consequence is that prompting introduces controlled variation. A detector that operates at the surface level (looking for specific words or patterns) can be fooled by prompts that instruct the model to write in a particular style. A detector that operates at deeper levels (analyzing probability distributions, syntactic structure, or attention patterns) may be more robust to prompting, because those deeper properties are determined by the model's weights rather than the input.

This is not absolute: a sufficiently powerful model can change even deep statistical properties in response to prompting. But in practice, many detectors find stable signals that persist across prompts. The key is to look for properties that are determined by the model's architecture and training rather than by the specific input.

These considerations show that text generation is not a single process but a family of processes, each with different parameters, different distributions, and different forensic signatures. Detection must account for this variation. The next two chapters develop specific detection techniques that do so, starting with statistical and linguistic methods and moving to neural and transformer-based approaches.

Chapter 3: Generative Image Models

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Generative Adversarial Networks: Training Dynamics and Artifacts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Variational Autoencoders and Latent Representations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Diffusion Models: Forward Process, Reverse Process, and Sampling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Score-Based Models and Noise Scheduling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Architectural Fingerprints and Latent-Space Structure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Image Quality Metrics and Their Forensic Relevance

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 4: Audio and Speech Synthesis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Audio Representation: Waveforms, Spectrograms, MFCCs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Waveform Synthesis: WaveNet, WaveRNN, and Autoregressive Models

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Vocoder Architectures: Griffin-Lim, Neural Vocoders, HiFi-GAN

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Text-to-Speech Pipelines and Prosodic Modeling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Voice Cloning and Speaker Adaptation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Singing and Music Generation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 5: Video Generation and Manipulation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Video Autoencoders and Temporal Modeling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Neural Rendering and 3D-Aware Generation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Face Swapping: AutoENCoders, StyleGAN-Based Methods, and SimSwap

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Lip-Sync and Talking-Head Synthesis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Temporal Consistency Challenges

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Video Compression Interaction with Generative Artifacts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 6: Digital Forensics and Signal Processing Fundamentals

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Digital Evidence: Acquisition, Preservation, and Chain of Custody

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Image Forensics Fundamentals: Error Level Analysis, Noise Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Frequency-Domain Analysis: DFT, DCT, Wavelet Transforms

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Statistical Analysis: Hypothesis Testing, Distribution Fitting, Anomaly Detection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Metadata Standards: EXIF, XMP, ID3, Forensic Metadata

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

File Format Analysis and Structural Artifacts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 7: Text Detection Part One: Statistical and Linguistic Signals

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Perplexity and Likelihood-Based Detection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Burstiness, Entropy, and N-gram Statistics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Syntactic Complexity and Sentence Structure Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Semantic Coherence and Topic Consistency

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Stylometric Features and Authorship Attribution

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Implementation: Statistical Text Detector

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 8: Text Detection Part Two: Machine Learning and Transformer-Based Methods

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Supervised Classifiers: SVMs, Random Forests on Text Features

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Fine-Tuned Transformer Detectors: Architecture and Training

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Zero-Shot and Few-Shot Detection with Prompt Engineering

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Embedding-Based Detection and Clustering Approaches

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Source Attribution: Which Model Generated This Text?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Implementation: Fine-Tuned Transformer Text Detector

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 9: Image Detection Part One: Classical and Statistical Forensics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Error Level Analysis and Compression Mismatch Detection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Noise Residual Analysis and PRNU

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Frequency-Domain Anomalies in Generated Images

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Edge and Contour Artifacts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Illumination and Shadow Consistency Checks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Implementation: Error Level Analysis and Noise Residual Detector

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 10: Image Detection Part Two: Deep Learning-Based Detection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

CNN-Based Classifiers for GAN-Generated Image Detection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Frequency-Domain CNN Features

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Patch-Based and Local Inconsistency Detection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Transformer-Based Image Detectors

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Diffusion-Specific Detection: Noise Fingerprints and Sampling Artifacts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Implementation: CNN-Based Image Generator Detector

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 11: Image Detection Part Three: Metadata, Provenance, and Watermarking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Metadata Analysis: EXIF, XMP, and File Signatures

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Image Editing History Reconstruction

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Invisible Watermarking: Spread-Spectrum and Steganographic Methods

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Robust Watermarking for Diffusion Models

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

C2PA and Content Credentials Standard

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Implementation: Metadata Forensics and Watermark Analysis Pipeline

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 12: Video Detection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Frame-Level Detection: Applying Image Detectors to Video

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Temporal Inconsistency Analysis: Flickering, Blending Artifacts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Facial Biometric Cues: Blinking, Eye Contact, Micro-Expressions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Audio-Video Sync and Lip-Sync Verification

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Deepfake-Specific Artifacts: Boundary Regions, Skin Tone Inconsistencies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Implementation: Frame-Level Plus Temporal Video Detector

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 13: Audio Detection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Spectral Analysis and Anomaly Detection in Synthetic Speech

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Prosodic and Temporal Pattern Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Voice Biometric Verification and Speaker Comparison

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Neural Artifact Detection: Vocoder Fingerprints and Spectral Residuals

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Zero-Shot Speaker Verification for Deepfake Detection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Implementation: Spectral Analysis Speech Deepfake Detector

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 14: Multimodal Detection and Cross-Modal Consistency

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Visual-Text Consistency: Does the Image Match the Caption?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Audio-Visual Consistency: Does the Speech Match the Video?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Multimodal Transformers for Joint Detection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Cross-Modal Retrieval and Embedding Consistency

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Exploiting Generative Model Limitations in Multimodal Tasks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Implementation: Multimodal Consistency Checker

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 15: Watermarking and Cryptographic Provenance

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Visible vs. Invisible Watermarking: Trade-offs and Applications

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Spread-Spectrum Watermarking for Images and Audio

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Perceptual Hashing and Content-Based Authentication

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Cryptographic Signatures and the C2PA Framework

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Content Credentials: Implementation and Limitations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Blockchain-Based Provenance and Its Practical Constraints

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 16: Source Attribution and Model Fingerprinting

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Architectural Fingerprints in Outputs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Training Data Leakage and Memorization Artifacts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Statistical Signature Analysis for Source Identification

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Embedding-Based Model Fingerprinting

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Challenges: Model Variants, Fine-Tuning, and Ensemble Generators

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Case Study: Distinguishing Between Major Image Generators

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 17: Evaluation Methodology

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Confusion Matrix Metrics: Accuracy, Precision, Recall, F1, ROC-AUC, PR-AUC

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Cost-Sensitive Evaluation: False Positives vs. False Negatives

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Calibration: Confidence Scores, Reliability Diagrams, Temperature Scaling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Robustness Evaluation Across Generators, Domains, and Transformations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Dataset Contamination, Benchmark Leakage, and Evaluation Integrity

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Implementation: Complete Detector Evaluation Pipeline

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 18: Adversarial Attacks and Robustness

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Transformation-Based Attacks: Compression, Resizing, Denoising, Cropping

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Text-Specific Attacks: Paraphrasing, Translation, Style Transfer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Adversarial Perturbations: Gradient-Based and Optimization-Based Attacks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Prompt Engineering as an Evasion Technique

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Defenses: Ensemble Methods, Frequency-Domain Robustness, Watermarking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Implementation: Adversarial Robustness Testing Framework

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 19: Limitations and Fundamental Constraints

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Distribution Shift and Generalization Failure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

The Fundamental Impossibility of Universal Detection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Degradation from Post-Processing and Editing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Human vs. AI Boundary Blurring with Advanced Models

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Legal, Ethical, and Societal Constraints on Detection Deployment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

What We Cannot Know: Fundamental Epistemic Limits

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 20: Real-World Forensic Workflows

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Evidence Acquisition and Preservation Procedures

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chain of Custody and Documentation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Triage and Automated Screening

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Manual Forensic Review and Multi-Method Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Confidence Reporting and Expert Testimony Considerations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Case Study: Investigating a Suspected AI-Generated Video

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 21: Tools, Datasets, Standards, and Ecosystem

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Open-Source Detection Tools and Libraries

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Major Datasets: Text, Image, Video, and Audio Benchmarks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Standards Organizations: IEEE, ISO, NIST Efforts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Commercial Detection Systems and Their Claims

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Research Communities and Preprint Culture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Reproducibility Crisis in Detection Research

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Chapter 22: Future Directions and Responsible Practice

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Emerging Generative Techniques and Their Detection Challenges

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Active Provenance: Watermarks, Credentials, and Policy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Human-AI Collaboration in Forensic Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Responsible Disclosure and Red-Teaming

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Policy Implications and Governance Frameworks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Concluding Synthesis: Detection as Practice, Not Product

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

Conclusion

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.

References

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/theaiforensicshandbook.