VerityNgn: A Multimodal Counter-Intelligence System for YouTube Video Verification

Authors: VerityNgn Research Team
Affiliation: VerityNgn Open Source Project
Date: October 23, 2025
Version: 1.0

Abstract

We present VerityNgn, a novel automated system for assessing the truthfulness of claims made in YouTube videos through multimodal AI analysis combined with counter-intelligence techniques. The system extracts claims through frame-by-frame video analysis at 1 FPS sampling rate using Google's Gemini 2.5 Flash multimodal LLM, verifies them against external sources, and employs a unique counter-intelligence subsystem that analyzes YouTube review videos and detects press release bias. Our probabilistic framework combines evidence quality weighting, source credibility assessment, and counter-intelligence adjustments to generate nuanced truthfulness scores. In evaluation across 50+ videos spanning health, finance, and technology domains, VerityNgn achieved 78% accuracy in identifying misleading claims when compared to manual expert review.

Keywords: video verification, multimodal analysis, fact-checking, counter-intelligence, probability distribution, truthfulness assessment, LLM

1. Introduction

1.1 Problem Statement

The proliferation of misleading information in video content represents a significant challenge for information ecosystems. Unlike text-based content, video misinformation combines multiple modalities (visual, audio, textual) that make traditional fact-checking approaches insufficient [1]. YouTube, with 2.7 billion users and 500 hours of video uploaded every minute [2], has become a primary vector for health misinformation, financial scams, and pseudoscientific claims [3, 4].

Current automated fact-checking systems exhibit three critical limitations:

1. **Modal Incompleteness:** Most systems analyze only text transcripts, missing visual demonstrations, on-screen graphics, and audio-visual context [5, 6]
2. **Counter-Intelligence Blind Spot:** Existing systems do not systematically search for or weight contradictory evidence from review videos, community responses, or debunking content [7]
3. **Binary Classification Limitation:** Truth is often probabilistic, not binary, yet most systems force claims into "true" or "false" categories without expressing uncertainty [8]

1.2 Our Contribution

We introduce VerityNgn, which addresses these limitations through:

1. **Aggressive Multimodal Analysis:** Frame-by-frame sampling at 1 FPS with 64K token context window, analyzing video, audio, OCR, and visual demonstrations simultaneously
2. **Novel Counter-Intelligence System:** Automated search and analysis of YouTube review videos and press release detection to identify contradictory evidence and promotional bias
3. **Probabilistic Truth Assessment:** Three-state probability distribution (TRUE, FALSE, UNCERTAIN) with transparent evidence weighting and normalization
4. **Transparent Methodology:** Full disclosure of probability calculation factors, evidence sources, and reasoning steps

1.3 Paper Organization

Section 2 reviews related work. Section 3 details the system architecture. Section 4 presents the multimodal analysis pipeline. Section 5 describes the counter-intelligence methodology. Section 6 explains the probability model. Section 7 presents evaluation results. Section 8 discusses limitations and future work.

---

2. Related Work

2.1 Video-Based Fact-Checking

Traditional fact-checking systems focus primarily on textual content. ClaimBuster [9] extracts check-worthy claims from text using NLP. Full Fact [10] combines human verification with automated monitoring. However, these systems struggle with video content due to modal complexity.

Recent work has begun addressing video:

- **Video Verification using BERT** [11]: Analyzes transcripts only, missing visual content
- **Multimodal Fact-Checking** [12]: Limited to still images, not video
- **FakeTalkerDetect** [13]: Focuses on deepfake detection, not claim verification

Our Distinction: VerityNgn performs comprehensive multimodal analysis of real video content (not deepfakes) with integrated counter-intelligence.

2.2 Multimodal Analysis with LLMs

The advent of multimodal LLMs has enabled new verification approaches:

- **GPT-4V** [14]: Image understanding capabilities
- **Gemini Pro Vision** [15]: Video understanding with temporal analysis
- **LLaVA** [16]: Open-source multimodal understanding

Our Innovation: We use Gemini 2.5 Flash's 64K context window to analyze entire videos at 1 FPS, not just samples.

2.3 Counter-Intelligence in Information Warfare

Counter-intelligence techniques from information security inform our approach:

- **Contradiction Detection** [17]: Identifying conflicting statements
- **Source Credibility Assessment** [18]: Evaluating information sources
- **Bias Detection** [19]: Identifying promotional content

Our Contribution: First application of systematic counter-intelligence to automated video fact-checking, including YouTube review analysis and press release bias detection.

2.4 Probabilistic Truth Assessment

Previous work has explored uncertainty in fact-checking:

- **Bayesian Fact-Checking** [20]: Probabilistic updates with new evidence
- **Uncertainty-Aware Claim Verification** [21]: Confidence scores
- **Evidential Deep Learning** [22]: Uncertainty quantification

Our Approach: Evidence-weighted three-state probability distribution with transparent normalization and human-interpretable thresholds.

---

3. System Architecture

3.1 Overview

VerityNgn follows a six-stage pipeline architecture:

```
Input: YouTube URL →
Stage 1: Video Download & Preprocessing →
Stage 2: Multimodal Claim Extraction →
Stage 3: Counter-Intelligence Gathering →
Stage 4: Evidence Collection & Classification →
Stage 5: Probability Calculation →
Stage 6: Report Generation →
Output: Truthfulness Report (HTML/JSON/MD)
```

Each stage is implemented as a LangGraph [23] node, enabling robust error handling and state management.

3.2 Technical Stack

- **Multimodal LLM:** Google Gemini 2.5 Flash via Vertex AI

- **Video Processing:** yt-dlp for download and metadata
- **Search:** Google Custom Search API
- **Workflow:** LangGraph for orchestration
- **Storage:** Local filesystem or Google Cloud Storage

3.3 Design Principles

1. **Modularity:** Each stage can be independently tested and improved
2. **Transparency:** All decisions and calculations are logged and explained
3. **Fallback Mechanisms:** Graceful degradation when services unavailable
4. **Auditability:** Complete evidence trail for every verdict

---

4. Multimodal Analysis Pipeline

4.1 Video Preprocessing

Given YouTube URL `y`, the system:

1. **Downloads video:** Using yt-dlp with format selection

   ```
   format_spec = 'bestvideo[height<=720]+bestaudio/best[height<=720]'
   ```

2. **Extracts metadata:** Title, description, upload date, view count, likes, channel info
3. **Obtains transcripts:** Prioritize manual subtitles, fallback to auto-generated

4.2 Frame-by-Frame Analysis

The multimodal analysis operates at **1 frame per second** (aggressive sampling):

Input to Gemini 2.5 Flash:

- Video file (YouTube URL or uploaded video)
- Sampling instruction: 1 FPS
- Context window: 64K tokens
- Temperature: 0.1 (low for consistency)

Analysis Dimensions:

1. **Visual Channel:**
   - On-screen text extraction (OCR)
   - Graphics and charts interpretation
   - Demonstration analysis

    - Product displays
2. **Audio Channel:**
    - Spoken statement transcription
    - Speaker identification
    - Timestamp alignment
    - Tone analysis
3. **Metadata Analysis:**
    - Title and description keywords
    - Channel credentials
    - Comment sentiment
    - View/like ratios

4.3 CRAAP-Based Claim Extraction

Claims are extracted using the CRAAP criteria [24]:

- **Currency:** Temporal relevance of information
- **Relevance:** Centrality to video's message
- **Authority:** Claimed credentials or expertise
- **Accuracy:** Verifiability with external sources
- **Purpose:** Intent (educational vs promotional)

Prompt Engineering:

```
CRITICAL INSTRUCTIONS:
- Sample video at 1 FRAME PER SECOND
- Extract 5-15 specific, verifiable claims
- Focus on SPOKEN WORDS, VISUAL TEXT, DEMONSTRATIONS
- Prioritize claims verifiable against external sources

CLAIM MIX REQUIREMENTS:
- 70% Scientific & Verifiable Claims
- 10% Speaker Credibility Claims
- 20% Other Verifiable Claims

AVOID:
- Vague motivational statements
- General advice without specifics
- Obvious facts
- Subjective opinions
```

4.4 Structured Output

Each claim is formatted as:

```
{
  "claim_text": "Dr. X has 20 years of experience in endocrinology",
  "timestamp": "02:15",
  "speaker": "Dr. X (Narrator)",
  "source_type": "spoken",
  "initial_assessment": "Verifiable credential claim"
}
```

Quality Metrics:

- Specificity score
- Verifiability index
- Relevance rating
- CRAAP compliance

4.5 Evaluation of Multimodal Analysis

We evaluated claim extraction quality on 30 videos:

Metric	Score
Claim Detection Recall	85%
Claim Precision (valid claims)	92%
Timestamp Accuracy (±5s)	89%
Speaker Identification Accuracy	81%
Visual Text Extraction Recall	78%

---

5. Counter-Intelligence Methodology

5.1 Motivation

Traditional fact-checking focuses on supporting evidence. However, **contradictory evidence** is equally important for truthfulness assessment. We introduce two counter-intelligence mechanisms:

1. **YouTube Review Analysis:** Finding and analyzing videos that contradict the original
2. **Press Release Detection:** Identifying self-promotional content masquerading as independent verification

5.2 YouTube Counter-Intelligence

5.2.1 Search Strategy

For video `V` with title `T` and primary keywords `K`, generate queries:

```
queries = [
    f"{T} scam review",
    f"{T} fake exposed",
    f"{T} debunk analysis",
    f"{K} + warning review",
    f"{product_name} doesn't work"
]
```

Rationale: Counter-perspectives often use keywords like "scam", "fake", "exposed" in titles.

5.2.2 Video Filtering

Retrieved videos are filtered:

```
def is_relevant_counter_intel(video, original_video):
    # Must be different channel
    if video.channel_id == original_video.channel_id:
        return False

    # Must have meaningful views
    if video.view_count < 1000:
        return False

    # Must have recent upload (within 2 years)
    if (now - video.upload_date).days > 730:
        return False

    return True
```

5.2.3 Transcript Analysis

For each counter-intel candidate, extract transcript and analyze:

Counter-Signals:

```
counter_phrases = [
    'scam', 'fake', 'fraud', 'lie', 'misleading',
    'doesn\'t work', 'waste of money', 'red flags',
    'warning', 'beware', 'exposed'
]
```

Supporting Signals:

```
supporting_phrases = [
    'works', 'effective', 'results', 'recommend',
    'good', 'success', 'legit'
]
```

Stance Calculation:

$$
\text{counter_ratio} = \frac{\text{counter_signals}}{\text{counter_signals} + \text{supporting_signals}}
$$

```
if counter_ratio > 0.7:
    stance = 'counter'
    confidence = min(0.95, 0.6 + (counter_ratio - 0.7) * 1.17)
elif counter_ratio < 0.3:
    stance = 'supporting'
    confidence = min(0.95, 0.6 + (0.3 - counter_ratio) * 1.17)
else:
    stance = 'neutral'
    confidence = 0.5
```

5.2.4 Credibility Weighting

Counter-intelligence evidence is weighted by:

$$
W_{yt} = \alpha \cdot \log(1 + \text{views}) + \beta \cdot \text{stance_confidence} + \gamma \cdot \text{channel_score}
$$

Where:

- α = 0.4 (view count weight)
- β = 0.4 (stance confidence weight)
- γ = 0.2 (channel credibility weight)

Implementation:

```
def calculate_youtube_validation_power(video):
    view_component = 0.4 * min(2.0, math.log10(1 + video.view_count / 10000))
    stance_component = 0.4 * video.stance_confidence
    channel_component = 0.2 * video.channel_credibility

    return view_component + stance_component + channel_component
```

5.3 Press Release Detection

5.3.1 Detection Method

Press releases are identified through:

Domain Matching:

```
PRESS_RELEASE_DOMAINS = [
    'prnewswire.com', 'businesswire.com', 'globenewswire.com',
    'marketwired.com', 'accesswire.com', 'prweb.com'
]
```

Content Pattern Matching:

```
def is_press_release(text, url):
    indicators = [
        'press release',
        'newswire',
        'for immediate release',
        'contact: pr@',
        'announced today'
    ]
    return any(domain in url for domain in PR_DOMAINS) or \
           sum(text.lower().count(ind) for ind in indicators) >= 2
```

5.3.2 Self-Referential Detection

Press releases referencing the original video are particularly problematic:

```
def is_self_referential(source, original_video):
    video_title = original_video.title.lower()
    channel_name = original_video.channel.lower()
    source_text = source.text.lower()

    # Direct video reference
    if video_title in source_text:
        return True

    # Channel reference
    if channel_name in source_text:
        return True

    # Domain match
    if original_video.channel_domain in source.url:
        return True


    return False
```

```
    return false
```

5.3.3 Bias Quantification

Press releases receive negative validation power:

```
validation_power = -0.5  # Base penalty

if is_self_referential:
    validation_power = -1.0  # Increased penalty
```

Rationale: Self-promotional content lacks independent verification and introduces systematic bias toward positive claims.

5.4 Counter-Intelligence Evaluation

Evaluated on 25 videos with known misleading claims:

Metric	Performance
YouTube CI Detection Rate	76%
Press Release Detection Precision	94%
Press Release Detection Recall	87%
False Positive Rate (legitimate sources)	3%
Impact on Accuracy (misleading videos)	+18%

Key Finding: Counter-intelligence significantly improves detection of misleading content, especially in health and finance domains.

---

6. Probability Distribution Model

6.1 Three-State Framework

Unlike binary classification, we model truthfulness as a probability distribution over three states:

[
P(\text{TRUE}) + P(\text{FALSE}) + P(\text{UNCERTAIN}) = 1.0
]

Justification:

- Many claims lack sufficient evidence for definitive verdict
- Acknowledging uncertainty prevents overconfident false positives/negatives

- Acknowledging uncertainty prevents overconfident false positives/negatives
- Enables nuanced communication to users

6.2 Base Distribution

Initial probabilities before evidence:

```
base_dist = {
    "TRUE": 0.3,
    "FALSE": 0.3,
    "UNCERTAIN": 0.4
}
```

Rationale: Start with high uncertainty, let evidence drive towards TRUE or FALSE.

6.3 Evidence-Based Adjustments

Five factors adjust the base distribution:

Factor 1: Evidence Coverage

$$
\text{coverage_score} = \frac{|E|}{\lceil |C| / 5 \rceil}
$$

Where:

- $|E|$ = number of evidence sources
- $|C|$ = claim word count
- Division by 5 represents expected sources per 5 words

```
if coverage_score > 1.0:
    boost = min(0.3, coverage_score * 0.15)
    P(TRUE) += boost
```

Factor 2: Independent Source Ratio

$$
\text{independent_ratio} = \frac{|E_{\text{independent}}|}{|E_{\text{total}}|}
$$

```
if independent_ratio > 0.5:
    boost = min(0.25, (independent_ratio - 0.5) * 0.5)
    P(TRUE) += boost
```

Justification: Independent sources provide unbiased validation

Justification: Independent sources provide unbiased validation.

Factor 3: Scientific Evidence Weight

$$
W_{\text{science}} = \min\left(0.4, \sum_{e \in E_{\text{scientific}}} v(e) \times 0.2\right)
$$

Where v(e) is validation power of evidence e.

```
P(TRUE) += W_science
```

Justification: Scientific sources (peer-reviewed journals, research institutions) have highest credibility.

Factor 4: YouTube Counter-Intelligence

$$
W_{\text{yt}} = \min\left(0.20, \sum_{y \in Y_{\text{counter}}} v(y) \times 0.08\right)
$$

```
P(FALSE) += W_yt
```

Note: Impact deliberately limited to 0.20 to avoid over-aggressive FALSE bias (see Section 7.3).

Factor 5: Press Release Penalty

$$
\text{penalty} = \min\left(0.4, |E_{\text{self-ref}}| \times 0.15\right)
$$

```
P(FALSE) += penalty
```

Justification: Self-promotional content systematically overestimates TRUE claims.

6.4 Normalization

After adjustments, ensure valid probability distribution:

```
total = P(TRUE) + P(FALSE) + P(UNCERTAIN)

# Normalize
P(TRUE) = P(TRUE) / total
P(FALSE) = P(FALSE) / total
P(UNCERTAIN) = P(UNCERTAIN) / total

# Ensure minimum thresholds (avoid 0 probabilities)

for state in [TRUE, FALSE, UNCERTAIN]:
```

```
    P(state) = max(0.001, P(state))

# Re-normalize if needed
total = P(TRUE) + P(FALSE) + P(UNCERTAIN)
for state in [TRUE, FALSE, UNCERTAIN]:
    P(state) = P(state) / total
```

6.5 Verdict Mapping

Continuous probabilities mapped to discrete verdicts using **enhanced thresholds** (relaxed from conservative defaults):

```
T = P(TRUE) * 100
F = P(FALSE) * 100
U = P(UNCERTAIN) * 100

# Combined thresholds for nuance
TU_combined = T + U
FU_combined = F + U

if T > 70 and F < 10:
    verdict = "HIGHLY_LIKELY_TRUE"
elif TU_combined > 65 and F < 35:
    verdict = "LIKELY_TRUE"
elif T > 40 and F < 35:
    verdict = "LEANING_TRUE"
elif abs(T - F) < 10:
    verdict = "UNCERTAIN"
elif F > 35 and T < 30:
    verdict = "LEANING_FALSE"
elif FU_combined > 65 and T < 35:
    verdict = "LIKELY_FALSE"
elif F > 75:
    verdict = "HIGHLY_LIKELY_FALSE"
```

Threshold Justification:

- 65% threshold for "likely" allows for meaningful uncertainty
- 70% threshold for "highly likely" maintains high confidence bar
- Combined (TRUE + UNCERTAIN) and (FALSE + UNCERTAIN) allow nuanced assessment
- 10% difference tolerance for UNCERTAIN maintains skepticism

6.6 Mathematical Properties

Theorem 1: The normalization procedure preserves the order of state probabilities.

Proof: Let $P_0 = (T_0, F_0, U_0)$ be pre-normalization probabilities with $T_0 > F_0$. After normalization:

Proof: Let $P_0 = (T_0, F_0, U_0)$ be pre-normalization probabilities with $T_0 > F_0$. After normalization:

$$
T = \frac{T_0}{T_0 + F_0 + U_0}, \quad F = \frac{F_0}{T_0 + F_0 + U_0}
$$

Since $T_0 > F_0$ and the denominator is constant:

$$
\frac{T_0}{T_0 + F_0 + U_0} > \frac{F_0}{T_0 + F_0 + U_0} \Rightarrow T > F
$$

∎

Corollary: Evidence adjustments maintain their intended directional impact after normalization.

---

7. Evaluation

7.1 Dataset

We constructed a test set of 50 YouTube videos:

Category	Count	Characteristics
Health Claims	15	Supplement advertisements, medical advice
Financial Advice	15	Investment schemes, crypto promotions
Product Reviews	10	Tech products, consumer goods
Scientific Education	10	Popular science channels

Ground Truth: Manual expert review by domain specialists (M.D., CFA, Ph.D.) with consensus labeling.

7.2 Metrics

1. **Accuracy:** Agreement with expert consensus verdict
2. **Precision/Recall:** For "misleading" class (FALSE verdicts)
3. **Calibration:** Alignment of probability scores with actual truthfulness
4. **Explanation Quality:** Human evaluation of generated explanations

7.3 Results

Overall Performance

Metric	Score	95% CI

Metric	Score	95% CI
Accuracy	78%	[72%, 84%]
Precision (misleading)	82%	[75%, 89%]
Recall (misleading)	73%	[65%, 81%]
F1 Score	0.77	[0.70, 0.84]

Performance by Category

Category	Accuracy	Notes
Health Claims	83%	High press release detection rate
Financial Advice	76%	Complex claims, mixed evidence
Product Reviews	75%	Subjective elements complicate
Scientific Education	85%	Clear factual claims

Calibration Analysis

We binned predictions by confidence and measured actual accuracy:

Predicted Confidence	Actual Accuracy	Sample Size
90-100%	91%	15 claims
80-90%	84%	23 claims
70-80%	76%	31 claims
60-70%	68%	28 claims
< 60%	55%	43 claims

Calibration Plot: Near-diagonal, indicating well-calibrated probabilities (Brier score = 0.18).

7.4 Ablation Studies

Impact of Counter-Intelligence

Configuration	Accuracy	Accuracy

Configuration	Accuracy	ΔAccuracy
Full System	78%	-
No YouTube CI	68%	-10%
No Press Release Detection	71%	-7%
No Counter-Intel (both)	60%	-18%

Key Finding: Counter-intelligence components provide substantial accuracy improvement, especially for misleading content.

Impact of Multimodal Analysis

Configuration	Claim Extraction Quality	Verification Accuracy
Full Multimodal (1 FPS)	92%	78%
Transcript Only	68%	65%
Keyframes Only (1 per 10s)	81%	71%

Key Finding: Dense frame sampling (1 FPS) significantly improves claim extraction quality.

7.5 Error Analysis

False Positives (Marked FALSE, Actually TRUE)

Case: "Vitamin D supplementation reduces risk of respiratory infections"

- **System Verdict:** LIKELY FALSE (67%)
- **Expert Verdict:** TRUE (supported by meta-analysis)
- **Root Cause:** Press release detection over-fired, multiple legitimate sources misclassified
- **Lesson:** Need better press release heuristics for established medical facts

False Negatives (Marked TRUE, Actually FALSE)

Case: "Our product has 15,000 5-star reviews on Amazon"

- **System Verdict:** LEANING TRUE (55%)
- **Expert Verdict:** FALSE (reviews were purchased)
- **Root Cause:** Lack of review authenticity checking
- **Lesson:** Need Amazon review verification integration

7.6 Comparison with Baselines

System	Accuracy	Explanation	Multimodal

System	Accuracy	Explanation	Multimodal
VerityNgn (Ours)	78%	✓	✓
GPT-4 + Manual Prompting	62%	Limited	✗
ClaimBuster [9]	54%	✗	✗
InVID [25]	48%	✗	Partial

Note: Direct comparison difficult due to different evaluation sets. Numbers indicate order-of-magnitude performance.

---

8. Discussion

8.1 Strengths

1. **Comprehensive Analysis:** First system to combine multimodal video analysis with counter-intelligence
2. **Transparency:** Full methodology disclosure enables reproducibility and critique
3. **Probabilistic Reasoning:** Acknowledges uncertainty rather than forcing binary verdicts
4. **Evidence Attribution:** Every verdict linked to specific sources with validation power
5. **Adaptability:** Modular design allows component upgrades without system overhaul

8.2 Limitations

8.1 Current Limitations

1. **Language:** English-only due to LLM training limitations
2. **Computational Cost:** 1 FPS sampling requires significant compute (~ 2-5 minutes per video minute)
3. **API Dependencies:** Reliance on Google services (Vertex AI, Custom Search)
4. **Context Boundaries:** 64K token limit restricts very long videos (>2 hours)
5. **Subjectivity:** Some claims inherently subjective, resist objective verification
6. **Temporal Lag:** Analysis is point-in-time, new evidence published later not included

8.2.2 Ethical Considerations

Automated Truthfulness Scoring Risks:

- Over-reliance on system without human critical thinking
- Potential for gaming (adversarial content designed to fool system)
- Censorship concerns if deployed for content moderation
- Bias amplification from training data and search results

Mitigation Strategies:

1. Display probabilities, not just binary verdicts
2. Provide evidence sources for user review
3. Include "UNCERTAIN" category for ambiguous cases
4. Explicit disclaimers about system limitations
5. Open-source methodology for community scrutiny

Usage Guidelines:

- System designed to augment, not replace, human judgment
- Users should review evidence themselves
- Not intended for automated content moderation
- Best used as a research tool and starting point for investigation

8.3 Future Work

Technical Improvements

1. **Multi-Language Support**
    - Integrate translation APIs for claim extraction
    - Cross-language evidence validation
    - Cultural context understanding
2. **Real-Time Monitoring**
    - Continuous evidence updates post-analysis
    - Alert system for new contradictory evidence
    - Temporal score tracking
3. **Enhanced Visual Analysis**
    - Deeper computer vision integration
    - Object detection and tracking
    - Visual manipulation detection (cheapfakes)
4. **Community Validation**
    - Crowd-sourced evidence submission
    - Expert reviewer network
    - Public challenge mechanism
5. **Causal Reasoning**
    - Logical consistency checking
    - Causal chain analysis
    - Contradiction detection across claims

Research Directions

1. **Adversarial Robustness**
    - Study of deliberate attempts to fool system
    - Adversarial training approaches
    - Robustness metrics
2. **Cross-Platform Analysis**
    - Twitter, TikTok, Instagram video analysis
    - Platform-specific counter-intelligence
    - Cross-platform claim tracking
3. **Explainability**
    - Visual probability explanations
    - Interactive evidence exploration
    - Attention visualization for multimodal analysis
4. **Long-Term Evaluation**
    - Longitudinal accuracy tracking
    - Evidence evolution over time
    - Claim persistence analysis

---

9. Conclusion

We have presented VerityNgn, a novel system for automated video verification that combines multimodal AI analysis with counter-intelligence techniques. Our key contributions include:

1. **Multimodal Analysis Pipeline:** Frame-by-frame video analysis at 1 FPS using large context window (64K tokens)
2. **Counter-Intelligence System:** Automated YouTube review analysis and press release bias detection
3. **Probabilistic Framework:** Evidence-weighted three-state probability distribution with transparent calculations
4. **Comprehensive Evaluation:** 78% accuracy on diverse video dataset with ablation studies
5. **Open Methodology:** Full transparency enabling reproducibility and community improvement

VerityNgn demonstrates that automated video verification is feasible with current multimodal LLM technology, but significant challenges remain. We believe transparency and acknowledgment of uncertainty are crucial for responsible deployment of such systems.

Open Source Release: All code, methodology documentation, and evaluation datasets are available at

[repository URL].

References

[1] Shu, K., et al. "Fake News Detection on Social Media: A Data Mining Perspective." ACM SIGKDD Explorations, 2017.

[2] YouTube Statistics. "YouTube by the Numbers: Stats, Demographics & Fun Facts." Omnicore Agency, 2024.

[3] Wang, Y., et al. "Systematic Literature Review on the Spread of Health-related Misinformation on Social Media." Social Science & Medicine, 2019.

[4] Pennycook, G., & Rand, D. G. "Fighting misinformation on social media using crowdsourced judgments of news source quality." PNAS, 2019.

[5] Popat, K., et al. "Credibility Assessment of Textual Claims on the Web." CIKM, 2016.

[6] Thorne, J., & Vlachos, A. "Automated Fact Checking: Task formulations, methods and future directions." arXiv:1806.07687, 2018.

[7] Nakamura, K., et al. "r/Fakeddit: A New Multimodal Benchmark Dataset for Fine-grained Fake News Detection." LREC, 2020.

[8] Augenstein, I., et al. "Multi Task Learning for Argumentation Mining in Low-Resource Settings." NAACL, 2018.

[9] Hassan, N., et al. "ClaimBuster: The First-ever End-to-end Fact-checking System." VLDB, 2017.

[10] Babakar, M., & Moy, W. "The State of Automated Factchecking." Full Fact, 2016.

[11] Khattar, D., et al. "MVAE: Multimodal Variational Autoencoder for Fake News Detection." WWW, 2019.

[12] Zlatkova, D., et al. "Fact-Checking Meets Fauxtography: Verifying Claims About Images." EMNLP, 2019.

[13] Yang, X., et al. "Detecting Audio-Visual-Text Inconsistencies in Video Deep Fakes." WACV, 2021.

[14] OpenAI. "GPT-4V(ision) System Card." OpenAI Technical Report, 2023.

[15] Google. "Gemini: A Family of Highly Capable Multimodal Models." Google Technical Report, 2023.

[16] Liu, H., et al. "Visual Instruction Tuning." NeurIPS, 2023.

[17] Thorne, J., et al. "FEVER: a large-scale dataset for Fact Extraction and VERification." NAACL, 2018.

[18] Popat, K., et al. "Where the Truth Lies: Explaining the Credibility of Emerging Claims on the Web and Social Media." WWW, 2017.

[19] Chen, Y., et al. "Automatic Detection of Fake News." COLING, 2015.

[20] Pasternack, J., & Roth, D. "Knowing What to Believe (when you already know something)." COLING, 2010.

[21] Stammbach, D., et al. "Uncertainty-Aware Fact Verification via Evidentiality." EACL, 2023.

[22] Sensoy, M., et al. "Evidential Deep Learning to Quantify Classification Uncertainty." NeurIPS, 2018.

[23] LangChain. "LangGraph Documentation." LangChain, 2024.

[24] Blakeslee, S. "The CRAAP Test." LOEX Quarterly, 2004.

[25] Teyssou, D., et al. "The InVID plug-in: web video verification on the browser." MMM, 2017.

Acknowledgments

We thank the open-source community for LangChain, yt-dlp, and supporting libraries. We thank Google for Vertex AI access. We thank domain experts who provided ground truth labels for evaluation.

Appendix A: Detailed Probability Calculations

A.1 Example Walkthrough

Claim: "Product X causes 15 pounds of weight loss in 30 days"

Evidence Collected:

- 3 press releases from product manufacturer
- 1 YouTube review video "Product X Scam Exposed" (450K views, counter stance)
- 2 independent blog reviews (mixed results)
- 0 scientific studies

Base Distribution:

```
P(TRUE) = 0.3
P(FALSE) = 0.3
P(UNCERTAIN) = 0.4
```

Factor 1 - Evidence Coverage:

```
claim_words = 11
evidence_count = 6
coverage_score = 6 / (11 / 5) = 2.73

boost = min(0.3, 2.73 * 0.15) = 0.3
P(TRUE) = 0.3 + 0.3 = 0.6
```

Factor 2 - Independent Ratio:

```
independent_sources = 2
```

```
total_sources = 6
independent_ratio = 2 / 6 = 0.33

# Ratio < 0.5, no boost applied
```

Factor 3 - Scientific Evidence:

```
scientific_power = 0
# No boost applied
```

Factor 4 - YouTube Counter-Intelligence:

```
youtube_power = 1.8  # Based on view count and stance confidence
youtube_impact = min(0.20, 1.8 * 0.08) = 0.144

P(FALSE) = 0.3 + 0.144 = 0.444
```

Factor 5 - Press Release Penalty:

```
self_ref_count = 3
penalty = min(0.4, 3 * 0.15) = 0.4

P(FALSE) = 0.444 + 0.4 = 0.844
```

Pre-Normalization:

```
P(TRUE) = 0.6
P(FALSE) = 0.844
P(UNCERTAIN) = 0.4
Total = 1.844
```

Normalization:

```
P(TRUE) = 0.6 / 1.844 = 0.325 (33%)
P(FALSE) = 0.844 / 1.844 = 0.458 (46%)
P(UNCERTAIN) = 0.4 / 1.844 = 0.217 (22%)
```

Verdict Mapping:

```
T = 33%, F = 46%, U = 22%
# F > 35 and T < 30? No (T = 33)
# F > 35 and T < 35? Yes
→ Verdict: "LEANING FALSE"
```

Appendix B: System Implementation Details

B.1 Hardware Requirements

Minimum:

- CPU: 4 cores
- RAM: 16 GB
- Storage: 100 GB
- Network: Stable broadband

Recommended:

- CPU: 8+ cores
- RAM: 32 GB
- Storage: 500 GB SSD
- GPU: Not required (Vertex AI cloud-based)

B.2 API Requirements

- Google Cloud Platform account
- Vertex AI API enabled
- Custom Search API key
- YouTube Data API v3 key (optional, for enhanced metadata)

B.3 Cost Estimation

Per video analysis (10-minute video):

Service	Cost
Vertex AI (Gemini 2.5 Flash)	$0.15 - $0.30
Custom Search API (30 queries)	$0.15
YouTube Data API	Free (under quota)
Storage	$0.001
Total per video	**~$0.30 - $0.45**

This research paper is part of the VerityNgn open-source project. For code, data, and updates, visit the project

repository.