ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov686

An LLM-Augmented Multi-Modal Forensic Reasoning Framework for Detecting Tampered and AI-Generated Digital Images and Digital Documents

K. Karthika M. E.[1]; Sriram K.[2]; Venkatesh R.[3]; Vignesh T.[4]; Vel Murugan M.[5]

[1]Assistant Professor, Department of AI & DS PSR Engineering College Sivakasi, Tamilnadu, India
[2]Department of AI & DS PSR Engineering College Sivakasi, Tamilnadu, India
[3]Department of AI & DS PSR Engineering College Sivakasi, Tamilnadu, India
[4]Department of AI & DS PSR Engineering College Sivakasi, Tamilnadu, India
[5]Department of AI & DS PSR Engineering College Sivakasi, Tamilnadu, India

Publication Date: 2026/11/21

Abstract: Tampering detection of images, and of AI-generated (GAN/diffusion-synthesized) images in particular, and forensic inconsistencies of PDF, OOXML, and legacy binary documents are important issues related to the evaluation of integrity, authenticity and trustworthiness of digital evidence. Traditional single-signal detectors provide a label lacking interpretable evidence trail and thus are useless when justification of the verdict is needed. This paper introduces a desktop forensic analysis software called ForensiX AI that performs analysis of image and document forensics. In the case of images, a ResNet18 classifier (11.18M parameters) is used along with three traditional forensic signals — Error Level Analysis, a combination of noise-residual and frequency-domain modules, and EXIF/metadata analysis. All of those signals are combined in a weighted authenticity score (0.40 CNN + 0.25 ELA + 0.20 noise/frequency + 0.15 metadata), whereas Grad-CAM attention mapping is run independently as an explainability tool. Three specialist language models (visual, metadata, and AI-pattern analysis) are used to reconcile the obtained evidence to a severity-graded verdict according to the severity_max rule, which operates under the precedence rule that guarantees that the displayed verdict can never be cleaner than the underlying evidence. The classifier achieves 86.69% accuracy on a held-out set of 4,523 images, a figure reported with an explicit caveat: this set was used to choose the final checkpoint for deployment and thus is not fully independent. In the case of documents, a separate pipeline takes PDF, OOXML, and legacy binary (CFBF/OLE2) files according to magic bytes and processes them through the corresponding families of modules. The obtained evidence is then reconciled under a four-tier verdict classifier with the raising precedence assertion. Finally, a cross-pipeline module is responsible for invoking the image classifier and Grad-CAM on embedded pictures inside documents; this feature is enabled by default and verified to be CLEAN on 66 real Microsoft Office documents after four false-positive corrections. The software is built using Python, PyTorch, OpenCV, Tkinter, and Ollama. Results are reported with explicit caveats at all times, including a specialist reliability defect and a since-repaired specialist prompt whose current high accuracy is flagged as a suspected metadata-leakage artifact.

Keywords: *Digital Image Forensics, Document Forensics, Tampered Image Detection, AI-Generated Image Detection, PDF/OOXML Forensics, Convolutional Neural Networks, Large Language Models, Consensus Reasoning, Explainable AI.*

How to Cite: K. Karthika M. E.; Sriram K.; Venkatesh R.; Vignesh T.; Vel Murugan M. (2026) An LLM-Augmented Multi-Modal Forensic Reasoning Framework for Detecting Tampered and AI-Generated Digital Images and Digital Documents. *International Journal of Innovative Science and Research Technology*, 11(11), 997-1006. https://doi.org/10.38124/ijisrt/26nov686

I. INTRODUCTION

However, thanks to the widespread availability of user-friendly tools to create images, as well as recent advancements in image generation by means of diffusion and GAN models, it is nowadays extremely easy to create fake content in the form of an image. The same applies to documents, with PDF, Office documents, and binary formats all being possible to edit or forge using some tools that will leave traces which can be recovered through forensic investigation. It becomes imperative that forensic analysts, journalists, or content moderators receive more tools that would not just point to potentially suspicious content, but also provide a reasonable explanation of why the content should be suspected of anything. However, binary classifiers are not very helpful in this case because they provide one simple

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov686

verdict – whether a document is a real one or a forgery. The lack of evidence that can be independently investigated leaves no room for trust [9].

The classical image forensics methods provide individually weak yet complementary indicators, and the classifier-only deep learning systems, although accurate in-distribution, are notorious for poor generalization capability and inability to provide any interpretation [9], [10]. To address the above issues, the present paper proposes to employ a ResNet18 CNN classifier combined with classical forensic signal extractors, a parallel document pipeline processing PDF, OOXML, and binary legacy documents, and three-specialist LLMs reasoning layer for generating verifiable, graded verdicts based on provided evidence. The proposed methodology allows performing consistent and accurate forensic analysis for both modalities and is experimentally validated in this paper using actual measured data.

II. LITERATURE SURVEY

ELA was proposed by Krawetz [1] as a method of detecting inconsistencies in localized JPEG compression; ELA is employed herein as one of the signal fusion techniques for forensics, but it is considered untrustworthy for PNG images without any JPEG compression history.

Lukáš et al. [2] applied sensor pattern noise (SPN/PRNU-like) techniques for camera fingerprinting and thus tampering detection, contributing to the noise-residual component of this study's hybrid noise/frequency signal module.

In their work, Frank et al. [3] demonstrated that GAN-produced images have unique frequency artifacts irrespective of the architecture and dataset, which affects the computation of frequency domain submetrics in the same module.

A benchmark database of spliced/copy-move tampered images with ground truth masks was made available by Dong, Wang and Tan [4] (CASIA v2), which is used here in combination with CIFAKE.

The CIFAKE benchmark by Bird and Lotfi [5] contains ground truth images and latent-diffusion-generated images along with the use of Grad-CAM for explanation, which serves to inspire this work's unified classifier for classic tampering and AI-generated images.

The ResNet architecture with skip connections was developed by He et al. [6] and forms the CNN backbone in this work (ResNet18).

Temperature scaling, originally presented by Guo et al. [7] to address confidence calibration issues of overconfident softmax distributions, is implemented (T = 1.5106), but has yet to be integrated into the inference pipeline, as discussed in section 3.3.

Selvaraju et al. [8] developed Grad-CAM for gradient-based visualization of CNNs, which is used here as an attention overlay after scoring of authenticity signals.

Verdoliva [9] reviewed the generalization issues of deep media forensics across their training distribution, inspiring this work's practice of presenting held-out accuracy with independence guarantees, instead of in-distribution validation accuracy.

Multimodal large language models as reasoning layers on top of visual and structured evidences were surveyed by Yin et al. [10], supporting the three-specialist LLM reconciliation layer in this work.

Table 1 Key Comparisons

Ref	Topic area	Contribution	Limitations
[1]	JPEG Compression Forensics	Error Level Analysis for localized tampering detection	Can't be used with PNGs and other non-JPEG images
[2]	Sensor Noise Forensics	Camera/device identification through PRNU-style noise residual method	Susceptible to recompression and denoising operations
[4]	Tampering Benchmark Data	CASIA v2 dataset for splicing and copy-move tampering ground truth	Domain-specific; poor diversity in generators for AI-generated images
[9]	Forensic Generalization	State-of-the-Art survey on limitations of deep detector generalization	Survey-based; no actual deep detector contribution

III. METHODOLOGY

The systematic method used in the development of the suggested forensic reasoning approach entails the use of a CNN classifier, classical image forensics, a parallel document pipeline, reasoning through LLMs, and consensus aggregation, as described below:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov686

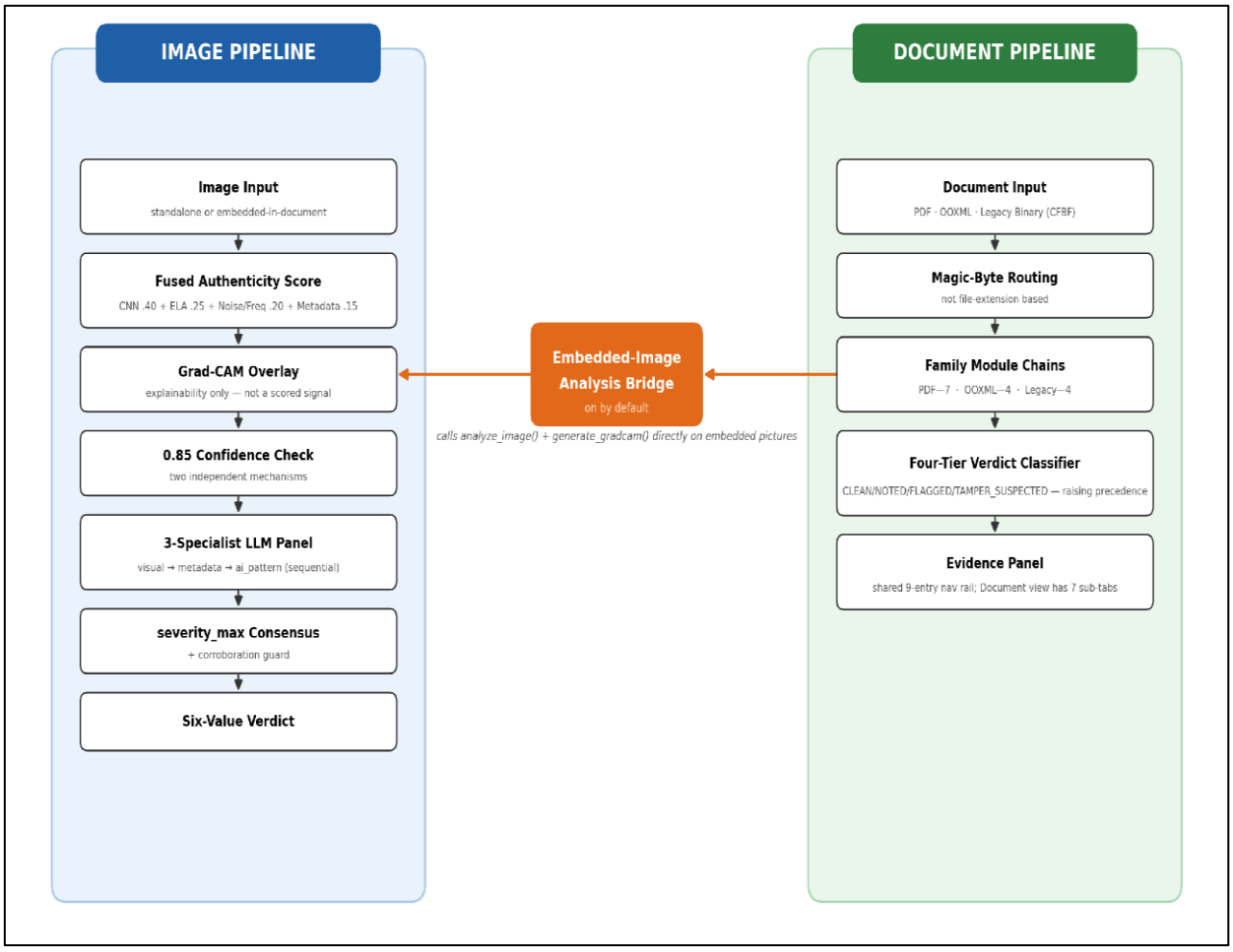


Fig 1 Proposed System Architecture

➢ *Image Acquisition and Preprocessing*

The system takes an input image or document supplied by the user through a GUI application using Tkinter and featuring drag and drop capability. The file type is identified using magic bytes in order to distinguish whether the file should pass through the image process flow or document process flow and not through file extensions.

➢ *Image Forensic Signal Fusion*

Instead of considering three classic forensic indicators individually as separate votes, they are combined with the CNN classifier into one weighted authentication score:

- Error Level Analysis (ELA) — weight 0.25: reveals inconsistencies in compression level, which signify localized tampering.
- Noise/Frequency Module – weight 0.20: one module calculating both the sensor-noise-residual metrics and the frequency periodicity metrics of GAN/diffusion synthesis; two analyses from one module, not two separate extractors.
- EXIF/metadata analysis – weight 0.15: examines camera and software metadata to check for either stripping or inconsistencies [11], using the POSSIBLE_WHATSAPP_RECOMPRESSION flag with lower penalty weight.
- CNN classifier score — weight 0.40: the biggest single factor, discussed below.

The Grad-CAM attention maps [8] are calculated independently, after the fusion score is generated, and are not weighted signals themselves but rather produce an attention heatmap for interpretability purposes.

➢ *CNN Classifier*

The main classifier employs a backbone of ResNet18 [6] (11,177,538 trainable parameters; input of 224×224 RGB images through simple resizing, without cropping) trained on a CIFAKE-CASIA v2 joint dataset, where the checkpoint with the highest validation accuracy (17/20 epochs) was chosen for deployment as the final model. Another checkpoint with additional data augmentation achieves better results in terms of best-validation accuracy but is not used for deployment (Section 5.1). Inference happens in a single-image, FP32 manner, without any autocasting. Temperature scaling [7] (T = 1.5106) was fitted for calibration purposes, but it is not used at the moment during inference for deployment (raw, uncalibrated softmax output). There are two different paths based on a 0.85 threshold for the softmax output of the above model: consensus source override (signal_trusted_high_cnn) and detection flag (CNN_HIGH_CONFIDENCE_FORGED).

➢ *Three-Specialist LLM Reasoning Layer*

Three specialist LLMs, independently prompted as visual_analyst, metadata_analyst, and ai_pattern_analyst, are used sequentially against locally hosted Ollama [15] models

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov686

(llava:7b, mistral:7b, gemma:7b), running under PyTorch [14] in order to accommodate an 8 GB VRAM limit. A consensus among the specialist outputs is achieved via the severity_max consensus strategy [12], whereby the severity of the most severe individual specialist evaluation decides the collective severity. There is a corroboration guard constraining when a specialist is able to escalate severity above the fused signal severity; the guard was extended during Phase 19 from guarding only escalations away from an AUTHENTIC signal reading, to guarding any escalation above the signal severity ranking — a strict superset, which, when tested offline using 256 cached evaluations, did not change a single binary accuracy result, and merely changed the classification of 16 rows from TAMPERED to SUSPICIOUS. Timeout or failure in a single specialist does not stop the whole process, but a partial specialist answer is *shown.*

➢ *Vocabulary and Consistency of Verdicts*

The image verdicts are calculated using a six-level severity vocabulary [13], wherein there is a unique derivation function generating the shown verdict for each surface (desktop UI, evidence panel, and CLI), so that the precedence principle (badge cannot possibly indicate higher cleaning level than the findings beneath it) is guaranteed to hold. The document pipeline uses a different and stricter four-level vocabulary (CLEAN / NOTED / FLAGGED_FOR_REVIEW / TAMPER_SUSPECTED, plus UNDETERMINED), with its own precedence check assertion that throws an error if violated — two related but different implementations, one per pipeline.

➢ *Document Forensic Pipeline*

The parallel pipeline examines PDF, OOXML (Word/Excel/PowerPoint), and legacy binary (CFBF/OLE2) documents using different module chains for each family, through a single analyze_document() API call separate from that for the image pipeline.

- PDF: incremental-update-chain analysis, metadata verification, font subset anomaly detection, embedded action analysis, page tree analysis, content stream analysis, embedded image analysis, and digital signature validation.
- OOXML: relationship graph analysis, metadata verification specific to OOXML, RSID discontinuity analysis, embedded image analysis, container analysis, metadata extraction, and revision analysis.
- Legacy binary (CFBF/OLE2): BIFF stream extraction, document content analysis, OLE stream analysis, and embedded image analysis, using common components for CFBF parsing and OLE property set parsing.

All three families use the same four-layer verdict classifier and metadata backend system; therefore, any findings in any family will be reported using the same severity vocabulary.

➢ *Cross-Pipeline Embedded-Image Analysis Module*

The module chain for each document family is completed with the embedded-image analysis module, which performs a direct invocation of the classifier and Grad-CAM generation from the image pipeline on the embedded images inside the document, under a two-axis applicability condition. The flags for image classifier confidence are converted into document severity labels – a high-confidence CNN forgery flag corresponds to TAMPER_SUSPECTED while a low-confidence one becomes NOTED – without changing the LLM panel of the document pipeline**.**

➢ *Training Dataset and Annotation*

- Containing real, altered, and AI-generated images (≈110,000 in total)
- CASIA v2 ground truth masks to label tampered regions
- Great diversity of tampering techniques, image sources, and compression histories

The test split has been fully recorded: 4,523 images (2,515 AUTHENTIC / 2,008 FORGED) – 2,000 from CIFAKE (1,000/class each) and 2,523 from CASIA v2 (1,515/1,008), seed 42. The number of images used for training and validation of the saved checkpoint have not been recorded in any of the remaining training logs and can only be calculated mathematically from the denominator of the validation accuracy of the checkpoint as follows: train: ~110,092 and validation: ~2.

➢ *Loss Functions and Calibration*

Training of the classifier involves the use of categorical cross entropy loss function,

$$L = -\Sigma\, y_i \log(p_i)$$

The temperature scaling coefficient is then learned after training and applied to the output of the softmax layer:

$$p_calibrated = softmax(z / T),\ T = 1.5106$$

However, this learned temperature scaling coefficient is not currently used in the deployed inference pipeline. The temperature T was also learned on the same 4,523 images split used in the held-out evaluation.

➢ *Inference Strategy*

- *Image Pipeline Steps:*

✓ Calculate fused authenticity score (CNN + ELA + frequency/noise + metadata)
✓ Create Grad-CAM mask independently (for explanation purposes only)
✓ Perform 0.85 high confidence override check
✓ Execute visual_analyst, metadata_analyst, and ai_pattern_analyst specialists one after another
✓ Aggregate specialist results using severity_max method (guarded by corroboration constraint)
✓ Use verdict precedence and create evidence panel

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov686

- *Document Pipeline Steps:*

✓ Redirect by magic bytes to the PDF, OOXML, or legacy binary module chain
✓ Execute the family specific module chain, including embedded image evaluation
✓ Classify using the common four-layer verdict classifier
✓ Apply the raising precedence rule and generate the Document evidence view

IV. IMPLEMENTATION

This implementation is a structured, modular one covering model specifications, system setup, training, and evaluation. This section outlines each step together with the exact configuration values and the numbers from the evaluation conducted by the project. Explicit caveats are attached whenever a number is not completely.

➢ *Model Input–Output Specification*

Table 2 Input–Output Specification

Parameter	Description
Input Image Size	224 × 224 RGB (resizing but no cropping)
Classes	2 (authentic, tampered/AI-generated)
Output	Probabilities for each class + confidence of raw softmax output

➢ *Model Configuration*

Table 3 Model Configuration Parameters

Parameter	Value
Backbone	ResNet18
Activation function	ReLU
Normalization	Batch normalization
Temperature scaling (T)	1.5106 (estimated; not used in inference yet)
Total number of parameters	11,177,538 trainable + 9,620 BN buffers
Parameter	Value

➢ *Training Hyperparameters*

Table 4 Training Hyperparameters

Hyperparameter	Value
Optimizer	AdamW
Initial Learning Rate	1e-4
LR Schedule	CosineAnnealingLR (eta_min=1e-6)
Batch Size	32
Best Checkpoint Epoch	17 of 20 (no early stopping)
Image Size	224 × 224
Weight Decay	1e-4
Loss	Cross-Entropy

Augmentation for the deployed checkpoint: Horizontal flip randomly, vertical flip randomly (probability=0.1), rotation randomly (±15°), color jitter (brightness/contrast 0.3, saturation 0.1), random grayscale conversion (probability=0.1), and random erasing.

➢ *System Configuration*

The implementation was done using Python, PyTorch, OpenCV, Tkinter, and Ollama for local LLM inference on the NVIDIA RTX 4060 Laptop GPU (8 GB VRAM), where there's a preflight check for VRAM usage that falls back automatically, ensuring specialist loading does not exceed budget. The Evidence GUI consists of a rail navigation menu with nine views: Dashboard, Verdict, Signals, Heatmap, Grad-CAM, Metadata, LLM Detail, Live Process, and Document, where the latter comes with seven sub-tabs. For reproducibility, we're using a fixed version of pypdf.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov686

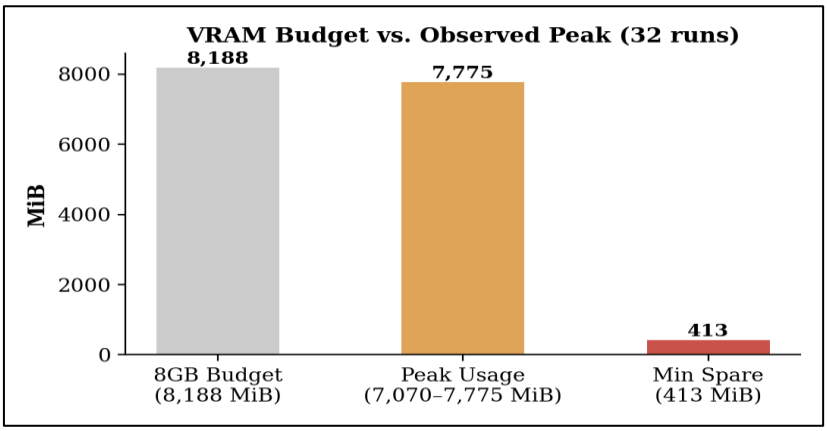


Fig 2 VRAM Budget vs. Observed Peak (32 Runs)

➢ *Classifier Performance*

On the 4,523-image held-out split, the deployed classifier (model.pt, epoch 17) achieves 86.69% accuracy (ROC-AUC 0.9418).

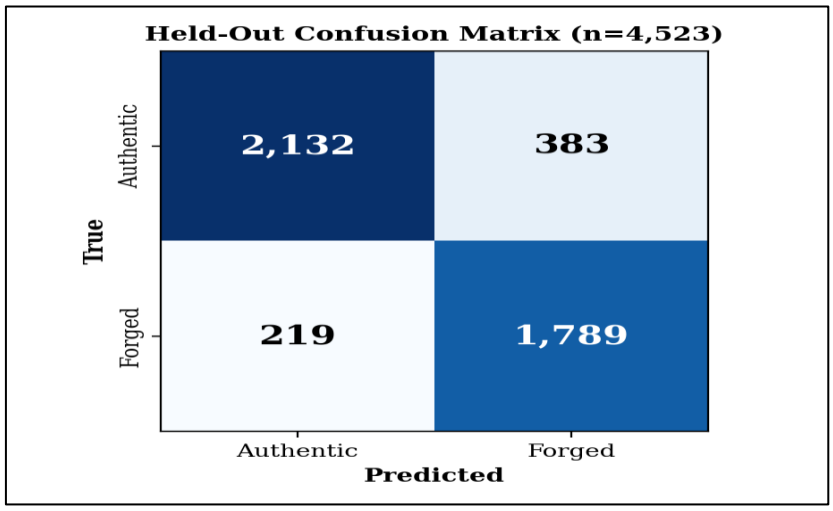


Fig 3 Held-Out Confusion Matrix (n=4,523)

There is one critical point to be made regarding this number: the held-out split used here also served as the criterion for model selection, so 86.69% is optimistic in nature and not a completely independent test estimate — clearly stated as "OPTIMISTIC" in the evaluation resources of the project itself. There is another checkpoint that was trained with more augmentation (and not used in deployment), which obtained 85.23% on the held-out split, which was truly not used in the training nor model selection process — this is the only completely independent held-out estimate.

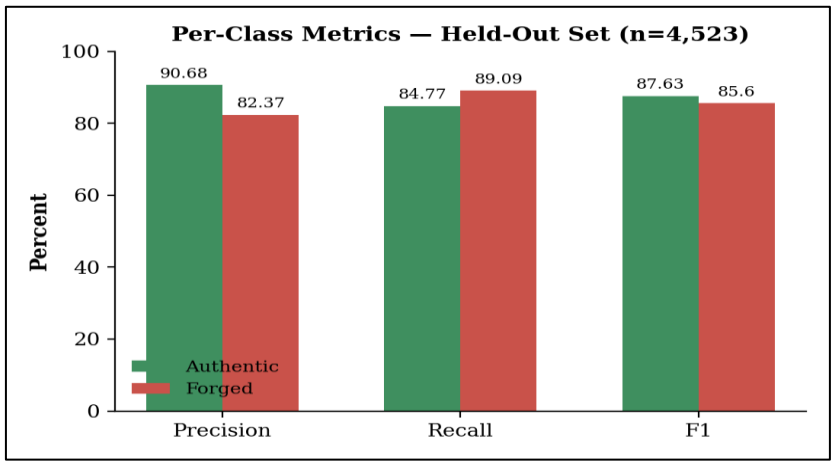


Fig 4 Per-Class Metrics — Held-Out Set (n=4,523)

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov686

➢ *System-Level Verdict Performance (Phase 19)*

For the Phase 19 held-out validation sweeps, the judgment output by the entire system, after collapsing it down to its binary Clean/Manipulated classification, was compared to ground truth in 14 out of 16 test images (the remaining 2 had no ground-truth labeling). There were no disagreements between the two sweeps.

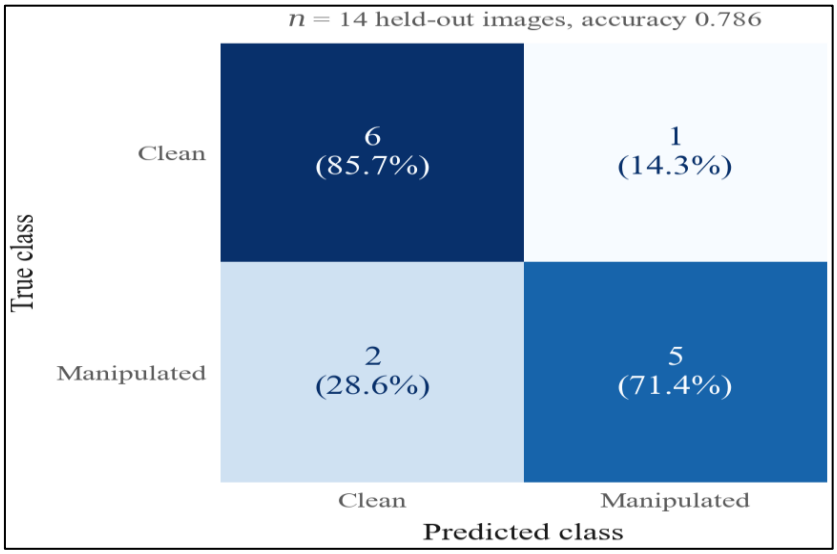


Fig 5 System Verdict Confusion Matrix (n=14) — Generated from Evaluation/phase19_Panel_Validation.Json

Derived metrics: precision (manipulated) 83.33%; recall/TPR 71.43%; specificity/TNR 85.71%; F1 (manipulated) 76.92%. This overall value is the end-to-end assessment of the entire system on a small hand-crafted stress test dataset, and should not be used to generalize about accuracy with n=14.

➢ *Specialist Ablation and Panel Configuration*

Ablation using n=128 (two disjoint samples) compared a metadata_analyst only configuration to the complete three specialist panel. The full panel did not outperform the metadata only configuration (both had 85.94% accuracy; McNemar p = 1.0), nor did the signal only configuration, which lacks the LLM layer entirely (85.16%).

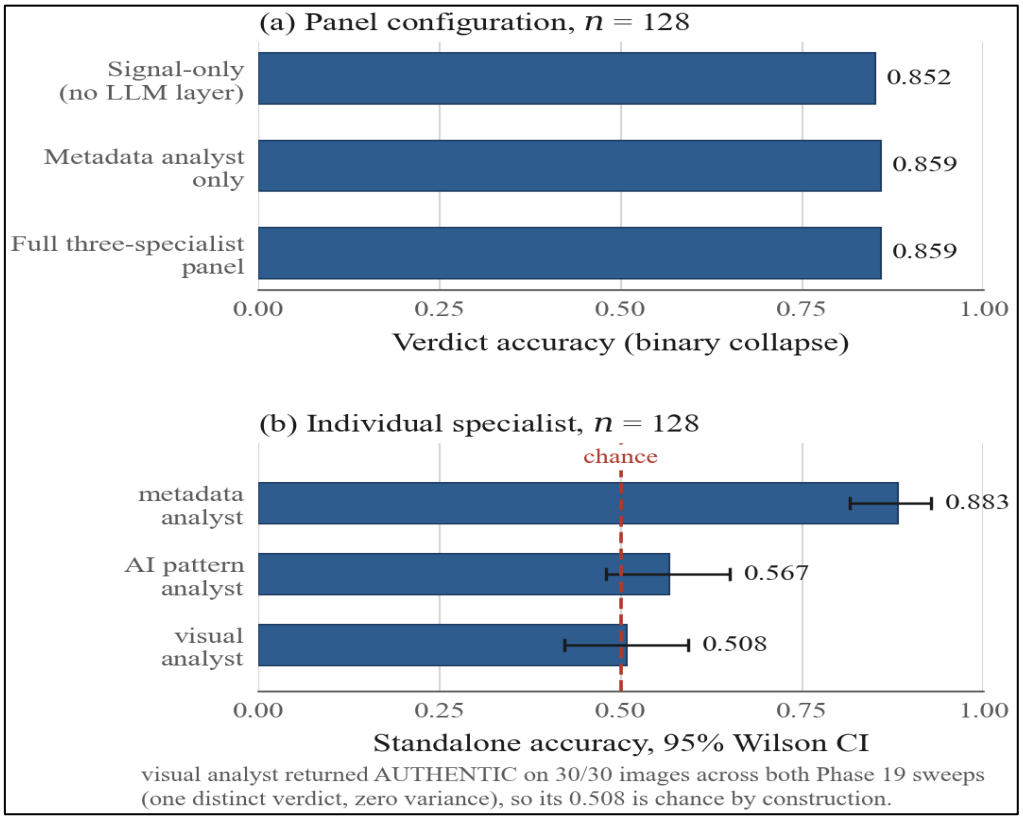


Fig 6 Panel Configuration and Per-Specialist Standalone Accuracy

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov686

This ablation corresponds to a prompt for ai_pattern_analyst that has been subsequently repaired via a Phase 16 patch. In terms of a single measurement made on the current shipping prompt, on a new n=64 draw: accuracy jumped to 93.8%, and the portion of votes making it the sole most-severe (decisive) increased from 9/128 to 36/64, and now alone it flips 23% of decisions (15/64), while previously flipping 0%. We don't present the 93.8% as verified forensic signal; it is internally marked as a potential CASIA/Photoshop metadata leakage.

➢ *Consensus Rule Comparison*

The comparison was done in Phase 11, not Phase 19. Severity_max performed at 84.38% compared to upper_median's 73.44% — exact McNemar test with two-sided alternative (b=8, c=1), p=0.0391. Earlier, in a non-independent fitted draw (Phase 10, n=64), the difference was even more pronounced: 87.5% vs. 70.3% (p=0.0034).

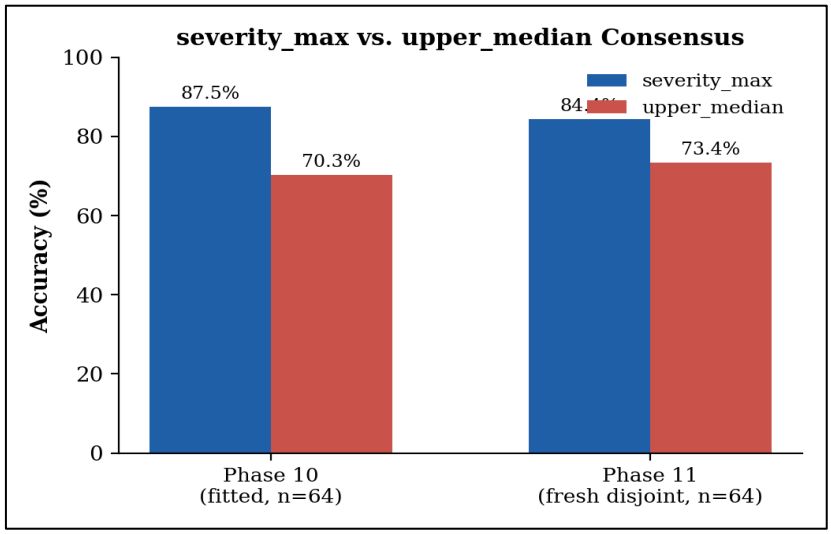


Fig 7 Severity_Max vs. Upper_Median Consensus Comparison

The rule was kept even after a bot-generated revert recommendation since upper_median performed worse out-of-sample in the new draw. In addition, the full three-specialist LLM layer failed to beat signal only configuration (both 84.38%, n=64, p=1.00) in the new draw evaluation.

➢ *Document Pipeline Validation*

The document pipeline was verified against 66 real-world Microsoft Office documents, all of which received a score of CLEAN after the identification of four false positive scenarios and correction of those issues: a ZIP entry predate document creation check (6 out of 6 real-world OOXML files), an isolated paragraph session heuristic (4 out of 8), a multiple subset tags for same font (11 out of 11 Word exported PDFs), and a prompt sent to a metadata analyst that incorrectly contained an image weight block within it. The synthetic fixture matrix for the tamper tier was unchanged at 54/54. A 12-fixture validation run against the real Ollama specialist in all three document families revealed no further gaps. None of the four false positives could be reproduced in the project's synthetic test fixtures - this became a methodological issue raised directly.

➢ *Qualitative Case: Visual_Analyst Failure Mode*

In both Phase 19 validation sweeps, visual_analyst registered an AUTHENTIC on all 30 of its 30 recorded votes — a constant classifier rather than a false negative outlier, even with respect to the two CASIA tampered/spliced images for which the system made an AUTHENTIC call (misses). The choice of weighting is undecided: visual_analyst continues to be weighted by SEVERITY_RANK at par with the two other specialists, and its absolute veto over any escalation from AUTHENTIC, if anything, becomes applicable to a wider variety of cases after the corroboration-guaranteed expansion of the scope of that rule — while the fundamental defect in reliability persists.

➢ *Evidence Panel and Dashboard*

The evidence panel is a nine-slot navigation rail (Dashboard, Verdict, Signals, Heatmap, Grad-CAM, Metadata, LLM Detail, Live Process, Document) implemented using Tkinter, defaulting to Dashboard. The Document view includes seven sub-tabs (Verdict, Signals, Metadata, LLM Detail, Embedded Images, Progress, Structure). All verdicts in use in the system UI and CLI come from a single function and are therefore guaranteed by design to respect the rule of precedence from Section 3.5.

➢ *Qualitative Evidence Examples (Grad-CAM and ELA)*

In addition to the statistics above, here are qualitative illustrations of both a legitimate strong point and a legitimate weakness in the current signal set. Fig 4.7 shows a Grad-CAM overlay on a real CASIA tampered composite, where the attention focus localizes accurately on the region of manipulation (rock/foreground); the heatmap is designed to do precisely this. On the other hand, Fig 4.8 shows a Grad-CAM analysis of a complete page snapshot (a sample invoice). Attention diffuses over the entire page rather than localizing on any particular region; we have seen this behavior in Grad-CAM analysis of other full pages, or large, text-dominant, and uniformly textured input in the validation set. We describe this as an honest weakness in attention

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov686

localization on the basis of gradients on document-style and/or low-contrast images, not as a success story. Fig 4.9 is a sample ELA analysis of CASIA spliced image, with the characteristic high-frequency speckle pattern used in ELA extraction visible in the compressed areas.

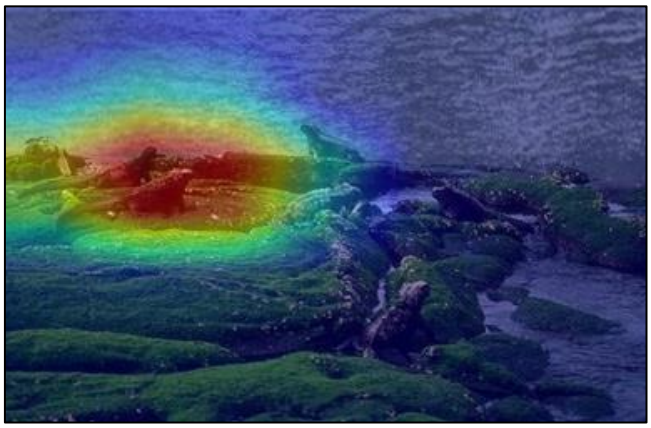


Fig 8 Grad-CAM — Localized Attention on a Tampered CASIA Sample

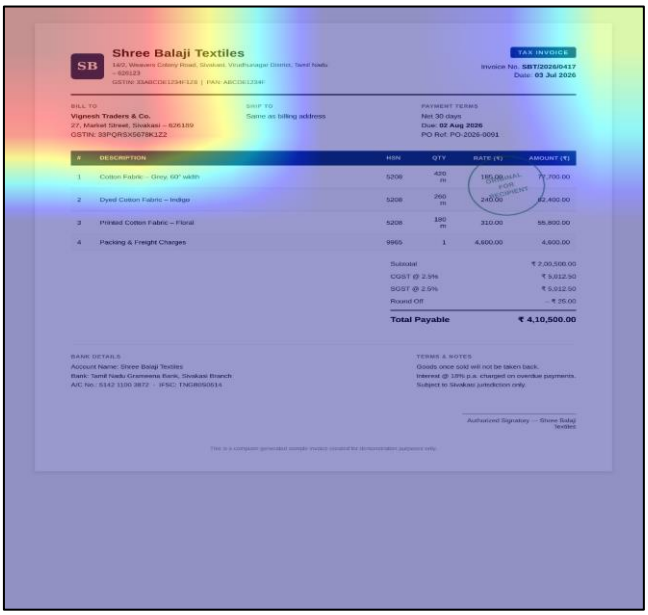


Fig 9 Grad-CAM — Diffuse, Non-Localized Attention on a Document Screenshot (Limitation)

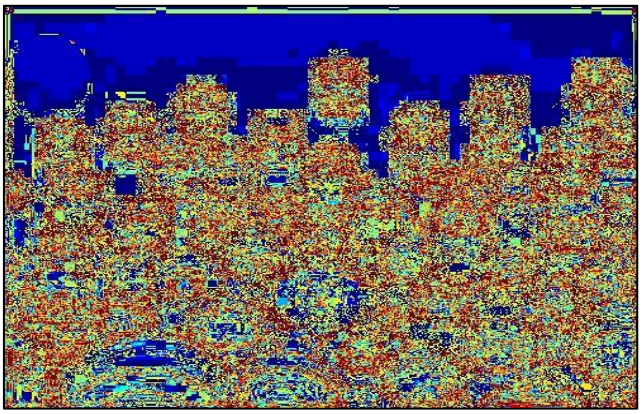


Fig 10 Error Level Analysis Output on a CASIA Splice Sample

V. RESULTS AND DISCUSSIONS

The system was assessed with regard to classifier performance, overall system-level verdict accuracy, specialist reliability, consensus-rule validity, and document pipeline correctness. Independent held-out classifier accuracy (86.69%) has an explicit non-independence caveat attached to it rather than being reported as a clean test metric; the only available fully independent held-out metric (85.23%) pertains to a checkpoint that does not ship. Overall system-level verdict accuracy on the small-phase-19 sweep (78.57%, n=14) is best interpreted as a stress test result, not a generalizable accuracy figure.

The most impactful unknown is the fix to the ai_pattern_analyst prompt: the ablation metrics that will be most frequently quoted from this system (Section 4.7) pertain to a superseded version of the prompt, and the current shipping version, which performs much better, has been explicitly called an artifact rather than verified signal - an outcome we think is more informative to report openly than to omit or report without proper caveats. Similarly, the new constant-AUTHENTIC output behavior of the visual_analyst specialist, together with the wider scope of the corroboration guard, imply that the impact of this specialist on final verdicts has actually increased, despite the unchanged reliability figure.

On the document side, the 66 document real file validation yielded four false positives that would never have shown up in any synthetic fixture in the test suite - underscoring the point that real file validation cannot be skipped for a forensic tool whose claims pertain to the real world. The cross-pipeline embedded image processing module implies that accuracy and calibration caveats pertaining to the image pipeline get propagated into the document pipeline whenever there are embedded images.

➢ *Future Enhancements*

Future work will involve connecting the temperature scaling parameter from the fit to the inference path and deriving a new 0.85 override threshold based on it; solving the visual_analyst specialist scoring conundrum (whether it should be removed from severity_max altogether, or whether the corroboration guard should go); compiling an out-of-distribution test set to stress-test generalization outside of the CIFAKE/CASIA distribution; and getting a genuinely independent held-out evaluation of the production checkpoint.

REFERENCES

[1.] N. Krawetz, "A Picture's Worth: Digital Image Analysis and Forensics," Black Hat Briefings USA, 2007.

[2.] J. Lukáš, J. Fridrich, and M. Goljan, "Digital Camera Identification from Sensor Pattern Noise," IEEE Trans. Inf. Forensics Security, vol. 1, no. 2, pp. 205–214, 2006.

[3.] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging Frequency

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov686

Analysis for Deep Fake Image Recognition," in Proc. ICML, 2020, pp. 3247–3258.
[4.] J. Dong, W. Wang, and T. Tan, "CASIA Image Tampering Detection Evaluation Database," in Proc. IEEE ChinaSIP, 2013, pp. 422–426.
[5.] J. J. Bird and A. Lotfi, "CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images," IEEE Access, vol. 12, pp. 15642–15650, 2024.
[6.] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE CVPR, 2016, pp. 770–778.
[7.] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in Proc. ICML, 2017, pp. 1321–1330.
[8.] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," in Proc. IEEE ICCV, 2017, pp. 618–626.
[9.] L. Verdoliva, "Media Forensics and DeepFakes: An Overview," IEEE J. Sel. Topics Signal Process., vol. 14, no. 5, pp. 910–932, 2020.
[10.] S. Yin, C. Fu, S. Zhao, K. Li, X. Sun, T. Xu, and E. Chen, "A Survey on Multimodal Large Language Models," arXiv preprint arXiv:2306.13549, 2023.
[11.] E. Kee, M. K. Johnson, and H. Farid, "Digital Image Authentication From JPEG Headers," IEEE Trans. Inf. Forensics Security, vol. 6, no. 3, pp. 1066–1075, 2011.
[12.] T. G. Dietterich, "Ensemble Methods in Machine Learning," in Multiple Classifier Systems (Lecture Notes in Computer Science, vol. 1857), Springer, 2000, pp. 1–15.
[13.] A. Barredo Arrieta et al., "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI," Information Fusion, vol. 58, pp. 82–115, 2020.
[14.] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Proc. NeurIPS, 2019.
[15.] Ollama, GitHub repository. [Online]. Available: https://github.com/ollama/ollama (accessed 2026).
[16.] H. Farid, "Exposing Digital Forgeries From JPEG Ghosts," IEEE Trans. Inf. Forensics Security, vol. 4, no. 1, pp. 154–160, 2009.
[17.] N. B. Abd Warif, A. W. A. Wahab, M. Y. I. Idris, R. Ramli, R. Salleh, S. Shamshirband, and K.-K. R. Choo, "Copy-Move Forgery Detection: Survey, Challenges and Future Directions," J. Netw. Comput. Appl., vol. 75, pp. 259–278, 2016.
[18.] Q. McNemar, "Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages," Psychometrika, vol. 12, no. 2, pp. 153–157, 1947.
[19.] E. B. Wilson, "Probable Inference, the Law of Succession, and Statistical Inference," J. Amer. Statist. Assoc., vol. 22, no. 158, pp. 209–212, 1927.
[20.] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual Instruction Tuning," in Proc. NeurIPS, vol. 36, pp. 34892–34916, 2023.
[21.] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al., "Mistral 7B," arXiv preprint arXiv:2310.06825, 2023.
[22.] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in Proc. ICLR, 2015.
[23.] Gemma Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, et al., "Gemma: Open Models Based on Gemini Research and Technology," arXiv preprint arXiv:2403.08295, 2024.
[24.] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Proc. NeurIPS, 2012, pp. 1097–1105.
[25.] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in Proc. ICML, 2015, pp. 448–456.
[26.] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014.
[27.] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in Proc. NeurIPS, 2017, pp. 5998–6008.
[28.] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in Proc. IEEE CVPR, 2009, pp. 248–255.
[29.] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, et al., "LLaMA: Open and Efficient Foundation Language Models," arXiv preprint arXiv:2302.13971, 2023.
[30.] OpenAI, "GPT-4 Technical Report," arXiv preprint arXiv:2303.08774, 2023.
[31.] T.-T. Ng and S.-F. Chang, "A Model for Image Splicing," in Proc. IEEE ICIP, 2004, pp. 1169–1172.
[32.] R. Salloum, Y. Ren, and C.-C. J. Kuo, "Image Splicing Localization Using a Multi-Task Fully Convolutional Network (MFCN)," J. Vis. Commun. Image Represent., vol. 51, pp. 201–209, 2018.
[33.] A. C. Popescu and H. Farid, "Exposing Digital Forgeries by Detecting Traces of Resampling," IEEE Trans. Signal Process., vol. 53, no. 2, pp. 758–767, 2005.
[34.] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to Detect Manipulated Facial Images," in Proc. IEEE ICCV, 2019, pp. 1–11.
[35.] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An Evaluation of Popular Copy-Move Forgery Detection Approaches," IEEE Trans. Inf. Forensics Security, vol. 7, no. 6, pp. 1841–1854, 2012.