System speed was another strength—complete verification (YOLO + OCR + NLP) took **less than 1 second**, making the system suitable for real-time applications.

5.5 Comparison & discussion

Compared to manual verification, the proposed system demonstrates significant advantages in terms of speed, accuracy, and reliability. Manual inspection often depends heavily on the experience and attention of the verifier, making it subjective and prone to human error—especially when dealing with subtle manipulations such as slight text edits, micro-forgery, or minor signature modifications. In contrast, the AI-powered approach performs consistently across all documents and can detect even pixel-level tampering that may go unnoticed during visual inspection. This automation reduces workload, eliminates human bias, and ensures uniform verification standards.

When compared to traditional OCR-based systems, the improvement is even more apparent. Conventional OCR tools focus solely on text extraction and lack the capability to analyze the actual document image for authenticity. They cannot identify forged seals, manipulated photos, or edited regions. The proposed system, however, integrates YOLOv8 to detect visual tampering, while OCR and NLP ensure that the extracted text is both correct and logically consistent. This dual-layer verification makes the system far more comprehensive than standard OCR tools.

Against commercial verification solutions, which are often expensive and limited to specific document types (such as passports or ID cards), the proposed system provides a cost-effective alternative. It offers greater flexibility by supporting a wide range of academic documents like certificates, mark sheets, and institutional IDs—formats that commercial tools often lack datasets for. Additionally, the modular design of the system allows institutions to customize the model according to their own document structures and verification requirements.

5.6 Experimental Setup

The experimental setup for the AI-Powered Document Verification System was carefully designed to ensure efficient training, accurate evaluation, and reliable execution of all components involved. All experiments were performed using **Google Colab Pro** because it provides free access to **GPU acceleration**, which is essential for training deep learning models such as YOLOv8. The system used a **NVIDIA T4 GPU (16GB VRAM)**, enabling faster training, real-time inference, and smooth processing of high-resolution document images. The programming environment was configured with **Python 3.10**, along with required libraries including OpenCV, NumPy, TensorFlow/PyTorch, and the Ultralytics YOLO framework.

The dataset used consisted of a combination of **real-world academic documents** (mark sheets, certificates, ID cards) and **synthetically forged samples** created manually using editing tools like Photoshop, Pixlr, and Photopea. These forged images simulated realistic tampering scenarios such as modified text, fake signatures, replaced photographs, altered seals, and background manipulation. The goal of this setup was to expose the model to diverse forgery patterns to enhance its generalization ability.

The backend API was tested using **FastAPI**, running on a local development environment and deployed temporarily on cloud-based servers such as **Render** for evaluation. The OCR component used **Gemini OCR API**, connected through internet calls, ensuring real-time extraction of textual fields. For NLP validation, the environment included **spaCy** and Regex tools to run field validation checks like date correctness, name format matching, and roll number validation.

5.7 Training / validation / test split

To ensure fair and balanced evaluation, the dataset was divided into three subsets:

- **70% Training Data:** Used to train YOLOv8 on forged and genuine document samples.
- **15% Validation Data:** Used during training to adjust model parameters and prevent overfitting.

- **15% Test Data:** Unseen data used to evaluate the model's final performance.

This split ensures that the system is trained on diverse examples but still tested on completely new documents to measure real-world performance.

5.8 Evaluation metrics

The performance of the system was measured using the following evaluation metrics:

Precision: Measures how many detected tampered regions were correctly identified.

Recall: Measures how many actual tampering regions the model successfully detected.

mAP (Mean Average Precision): Standard object detection score for YOLO models.

OCR Accuracy: Measures how accurately text fields were extracted from documents.

F1-Score: Balances precision and recall for overall performance.

Processing Time: Measures how fast the system verifies each document. These metrics help evaluate both vision-based forgery detection and text-based validation.

5.9 Results & Analysis

The YOLOv8 tampering detection model produced strong results, with:

- **Precision:** ~92%
- **Recall:** ~89%
- **mAP50:** ~94%

This shows the model was able to detect most manipulated areas accurately.
The **Gemini OCR module** achieved about **91% accuracy**, with minor errors in noisy or low-quality scans. NLP validation also performed well, successfully detecting incorrect formats, wrong dates, and inconsistent text fields.

System speed was another strength—complete verification (YOLO + OCR + NLP) took **less than 1 second**, making the system suitable for real-time applications.

5.10 Comparison & discussion

Compared to manual verification, the proposed system demonstrates significant advantages in terms of speed, accuracy, and reliability. Manual inspection often depends heavily on the experience and attention of the verifier, making it subjective and prone to human error—especially when dealing with subtle manipulations such as slight text edits, micro-forgery, or minor signature modifications. In contrast, the AI-powered approach performs consistently across all documents and can detect even pixel-level tampering that may go unnoticed during visual inspection. This automation reduces workload, eliminates human bias, and ensures uniform verification standards.

When compared to traditional OCR-based systems, the improvement is even more apparent. Conventional OCR tools focus solely on text extraction and lack the capability to analyze the actual document image for authenticity. They cannot identify forged seals, manipulated photos, or edited regions. The proposed system, however, integrates YOLOv8 to detect visual tampering, while OCR and NLP ensure that the extracted text is both correct and logically consistent. This dual-layer verification makes the system far more comprehensive than standard OCR tools.

Against commercial verification solutions, which are often expensive and limited to specific document types (such as passports or ID cards), the proposed system provides a cost-effective alternative. It offers greater flexibility by supporting a wide range of academic documents like certificates, mark sheets, and institutional IDs—formats that commercial tools often lack datasets for. Additionally, the modular design of the system allows institutions to customize the model according to their own document structures and verification requirements.

CHAP 6 : CONCLUSION AND FUTURE WORK

Conclusion

The AI-Powered Document Verification System developed in this project successfully demonstrates how modern artificial intelligence techniques can significantly enhance the accuracy, speed, and reliability of document authentication. By integrating **YOLOv8 for tampering detection**, **Gemini OCR for text extraction**, and **NLP-based validation** for consistency checking, the system provides a multi-layered verification pipeline that goes beyond traditional manual or OCR-only methods. The experimental results show strong performance in identifying forged regions, detecting modified text, recognizing altered signatures, and highlighting inconsistencies within academic documents such as mark sheets and certificates.

Unlike manual verification—which is slow, inconsistent, and prone to human error—the proposed system ensures standardized verification across all documents. It also surpasses conventional OCR systems by performing both **image-level analysis** and **text-level analysis**, making it suitable for modern digital environments. Overall, the system proves to be a practical, efficient, and scalable solution for educational institutions, HR departments, financial organizations, and government agencies that require secure and automated document authentication.

Future Work

Although the system performs well, there are several opportunities for future enhancements. First, expanding the dataset with more diverse document types such as passports, driving licenses, property documents, and bank statements would further improve the model's generalization ability. Additionally, integrating **advanced transformer-based models** for both vision (e.g., Vision Transformers) and text understanding (e.g., BERT) could enhance detection capability and reduce false positives.

Another potential improvement is implementing a **live document capture system** with camera-based preprocessing to prevent the upload of digitally altered images. Features like **liveness detection**, **watermark verification**, and **QR code integrity checking** can further strengthen security. Deploying the system on **edge devices** or creating a mobile application would make the verification process more accessible and convenient for end users. Finally, integrating an audit log and dashboard for administrators may provide

CHAP 7 : REFRENCES

[1] Kaggle, “Document Image Forgery Detection Dataset,” 2020. [Online]. Available: https://www.kaggle.com

[2] Ultralytics, “YOLOv8: State-of-the-Art Object Detection Model,” 2023. [Online]. Available: https://docs.ultralytics.com

[3] Google, “Gemini OCR API Documentation,” Google AI, 2024. [Online]. Available: https://ai.google.dev

[4] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” *arXiv preprint arXiv:2004.10934*, 2020.

[5] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[6] M. Abadi *et al.*, “TensorFlow: A System for Large-Scale Machine Learning,” in *Proc. 12th USENIX Symp. Operating Systems Design and Implementation*, 2016.

[7] A. Paszke *et al.*, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[8] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning Representations by Back-Propagating Errors,” *Nature*, vol. 323, pp. 533–536, 1986.

[10] R. Smith, “An Overview of the Tesseract OCR Engine,” in *Proc. 9th Int. Conf. Document Analysis and Recognition (ICDAR)*, 2007, pp. 629–633.

[11] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] NIST, “Special Database 19: Handwritten Forms and Characters,” National Institute of Standards and Technology, 2015.

[13] S. Prasad and A. Tripathi, “A Hybrid Deep Learning Approach for Document Forgery Detection,” *IEEE Access*, vol. 9, pp. 115204–115217, 2021.

[5] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded-Up Robust Features,” in *Proc. European Conf. Computer Vision (ECCV)*, 2006, pp. 404–417.