FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

Fake/Real Logo Detection using Deep learning

Functional Requirements:

1. Image Input:

- The system should accept images in standard formats (JPEG, PNG) with support for varying resolutions (minimum and maximum pixel dimensions).
- It should be able to handle images with different aspect ratios.

2. Logo Detection:

- Utilize deep learning algorithms for accurate logo detection.
- Implement techniques to handle occluded logos or logos with partial visibility.
- Support the detection of logos in cluttered or complex backgrounds.

3. Classification:

- Develop a robust classification model to distinguish between real and fake logos.
- Provide a confidence score for the classification to convey the model's level of certainty.

4. Training Data:

- The system should allow the integration of a diverse dataset containing real ansd fake logos.
- Enable periodic updates to the training dataset to improve the model's performance over time.

5. Accuracy:

- Define a target accuracy rate for both real and fake logo detection.

- Establish acceptable false positive and false negative rates based on application requirements.

6. Scalability:

- Design the system to handle an increasing number of logos and images efficiently.

- Implement distributed computing or parallel processing for scalability.

7. Integration:

- Provide RESTful APIs or SDKs for seamless integration with other systems.

- Support integration with popular programming languages and frameworks.

8. Real-time Processing:

- Ensure the system can process images in real-time or within an acceptable time frame for the specific application.

9. User Interface:

- If applicable, design an intuitive user interface displaying results, confidence scores, and allowing user interaction.

- Include features for manual verification and correction.

10. Logging and Reporting:

- Implement comprehensive logging of system activities for auditing.

- Generate reports on model performance, including accuracy metrics and processing times.

Non-Functional Requirements:

1. Performance:

- Achieve low-latency processing to meet real-time or near-real-time requirements.
- Define maximum response times for different levels of system load.

2. Reliability:

- Aim for high system availability with a defined acceptable downtime window for maintenance.
- Implement fault tolerance mechanisms to ensure continued operation in case of component failures.

3. Scalability:

- Design the system architecture to scale horizontally to accommodate increased processing demands.
- Define scalability benchmarks and conduct stress testing.

4. Security:

- Implement encryption for data in transit and at rest.
- Address potential security risks associated with the use of deep learning models, such as adversarial attacks.

5. Usability:

- Conduct usability testing to ensure the user interface is intuitive and user-friendly.

- Provide documentation for users, including guides for integration and troubleshooting.

6. Maintainability:

- Design modular and well-documented code for ease of maintenance.

- Implement version control for both the model and the software components.

7. Ethical Considerations:

- Address potential biases in the training data and model predictions.

- Implement mechanisms to monitor and mitigate ethical concerns, ensuring fairness and transparency.

8. Compliance:

- Ensure compliance with relevant data protection regulations (e.g., GDPR, HIPAA).

- Adhere to ethical guidelines and standards for AI and machine learning.

9. Resource Utilization:

- Optimize resource usage to minimize hardware requirements and energy consumption.

- Monitor resource utilization and conduct periodic performance audits.

10. Cross-Platform Compatibility:

- Ensure compatibility with various operating systems, browsers, and hardware configurations to maximize accessibility.