رواد مصر الرقمية

وزارة الاتصالات
وتكنولوجيا المعلومات

AI-GENERATED IMAGES DETECTION

DEPI Round 2 - Capstone Project

aws Machine Learning Track

رواد مصر الرقمية

وزارة الاتصالات وتكنولوجيا المعلومات

AGENDA

- Introduction
- Objectives
- Methodology
  - Data Collection & Preprocessing
  - Model Development & Training
  - Web Backend & Deployment
- Future Work & Limitations
- Conclusion

رواد مصر الرقمية

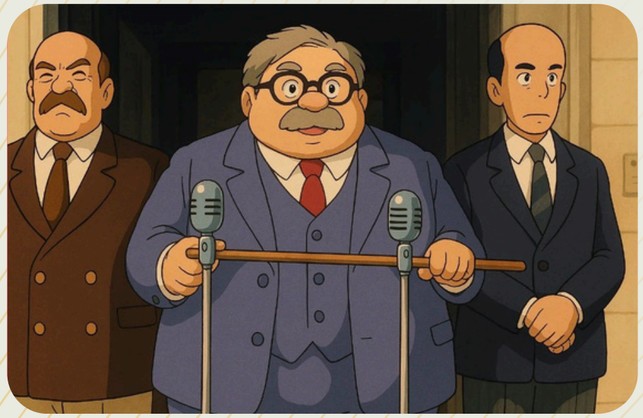
وزارة الاتصـــــالات
وتكنولوجيا المعلومات

INTRODUCTION

The line between real and AI-generated content is blurring fast. With tools like DALL·E, Midjourney now accessible to everyone, malicious use cases from deepfake propaganda to fake historical imagery are on the rise.

There’s a critical need for automated, scalable systems that can reliably detect such synthetic content. Manual verification doesn’t scale, and conventional tools fail to keep up with the realism of new AI models.

In this project, we are trying to respond to that need by developing a robust AI-powered image detection system that classifies content as either AI-generated or human-created.



رواد مصر الرقمية

وزارة الاتصـــــالات
وتكنولوجيا المعلومات

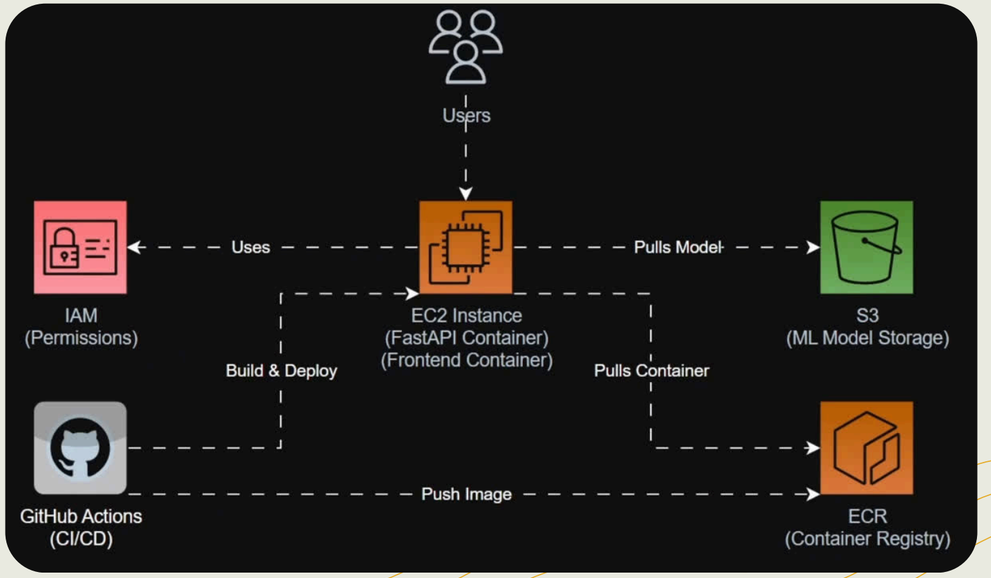
OBJECTIVES

Data Pipeline and Augmentation

Preprocess images at multiple resolutions and, apply robust augmentation strategies. To generalize the model

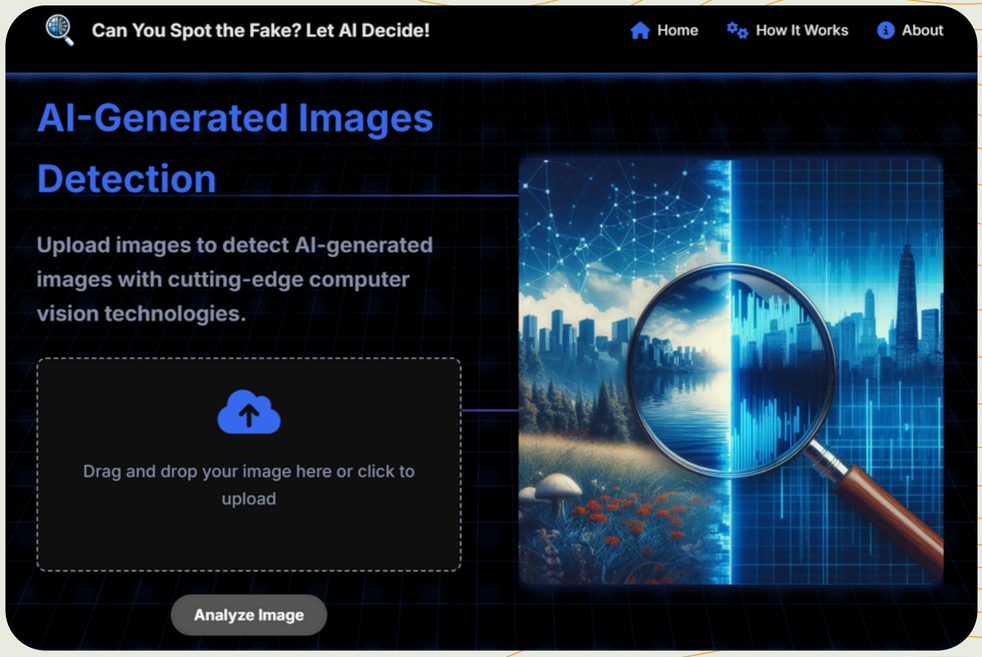
Model Development

Develop and compare state-of-the-art deep learning models for image classification, including variants from the EfficientNet and ConvNeXt families.



Web Backend & Deployment

- Create a FastAPI backend for image uploads and inference.
- Containerize the application using Docker and Docker Compose.
- Store models on AWS S3, push containers to AWS ECR, and run on AWS EC2.
- Automate deployments with GitHub Actions for CI/CD.



METHODOLOGY

1 - DATA COLLECTION & PREPROCESSING

Dataset: The dataset is sourced from the AI-Generated vs. Human Image Competition. The dataset for this Competition is provided by Shutterstock and DeepMedia, combines authentic and AI-generated images (Roughly 80,000 images) to create a robust foundation for training and evaluation.

AI-Generated Image Samples

Human-Created Image Samples

1 - DATA COLLECTION & PREPROCESSING

Preprocessing & Augmentations

Images are resized to resolutions (224 or 384) and normalized using pre-trained model-specific functions.
The augmentation pipeline uses Keras layers to apply transformations such as random cropping, flipping, rotation, translation, zoom, subtle blurring, brightness-color shifts.

AI-Generated Augmented Image Samples

Human-Generated Augmented Image Samples

1 - DATA COLLECTION & PREPROCESSING

Data Splitting: The dataset is split into 90% training (about 72,000 Images) and 10% validation (about 8,000 Images), with the Kaggle competition test set (about 5500 Images) used for final testing.

Competition Test set Samples

Test Image 1

Test Image 2

Test Image 3

Test Image 4

Test Image 5

Test Image 6

Test Image 7

Test Image 8

Test Image 9

Test Image 10

2 - MODEL DEVELOPMENT & TRAINING

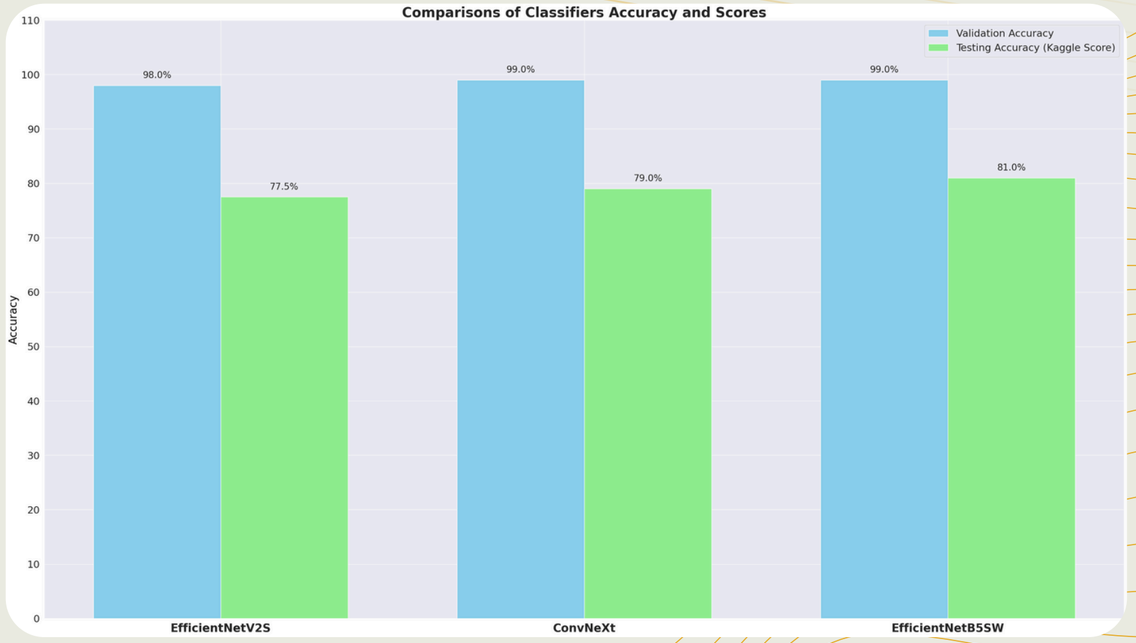
- **We experimented with multiple high-performing models:**

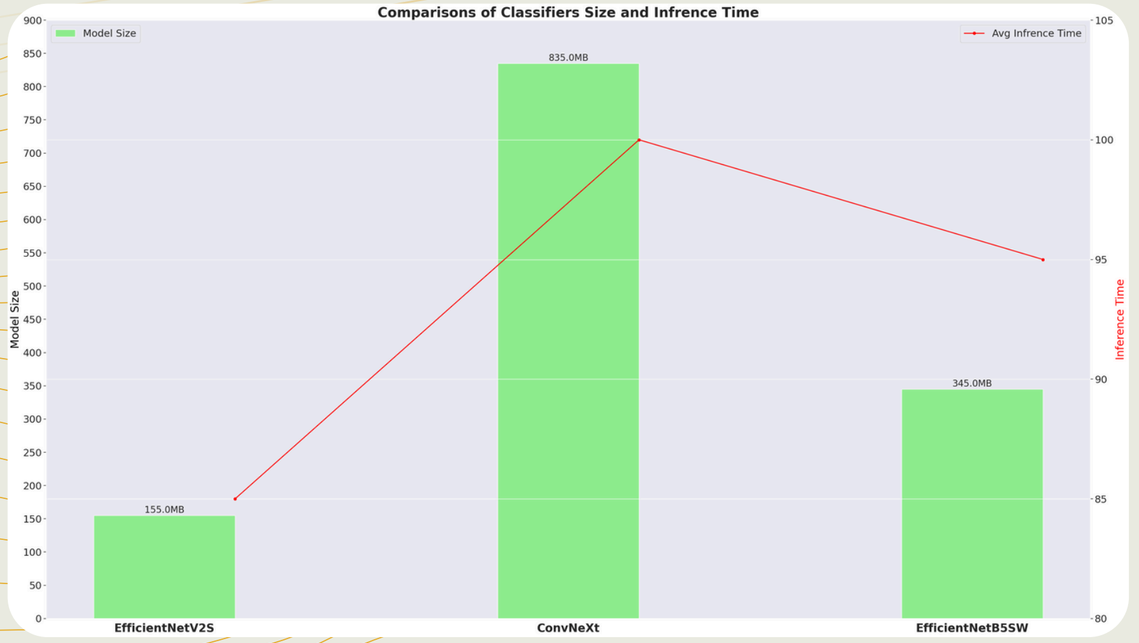
EfficientNetV2S
- Validation Score: 94%
- Kaggle Score: 77.5%

ConvNeXtTiny
- Validation Score: 96%
- Kaggle Score: 79%

EfficientNetVB5Swin
- Validation Score: 99%
- Kaggle Score: 81%

EfficientNetB5Swin was ultimately chosen for deployment due to its superior performance and efficient trade-off between accuracy, size, and inference speed—an essential trade-off for scalable, cloud-based deployment.





2 - MODEL DEVELOPMENT & TRAINING

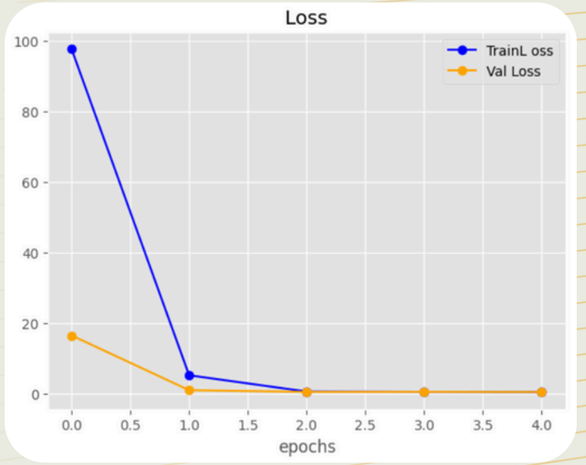
Training Configuration:

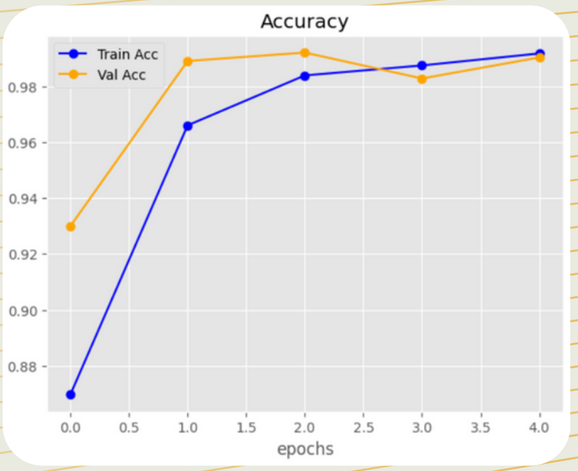
- **Optimizer:** AdamW
- **Training Duration:** 3–5 epochs
- **Loss Function:** Custom loss to address models bias towards a particular class in training.
- **Explanation:**
  - Loss 1: Standard Binary Cross-Entropy for per-sample classification.
  - Loss 2: Fairness Penalty using Mean Squared Error (MSE) to align predictions with a target class ratio (β).

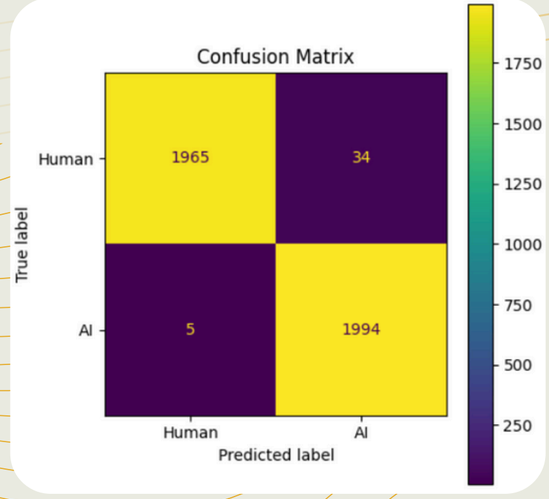
$$MSE = (mean\,(y_{pred}) - \beta)^2$$

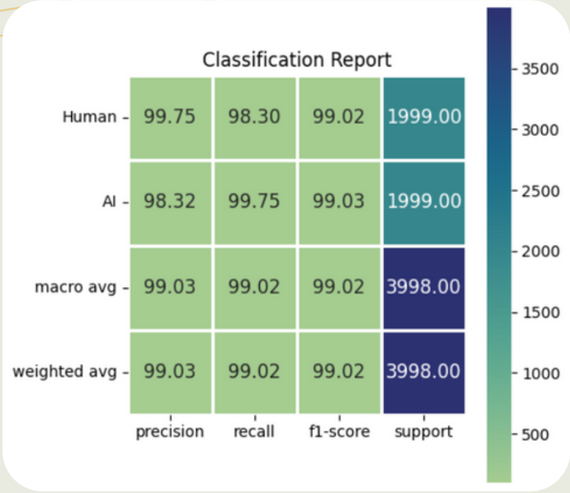
$$TotalLoss = Loss_1 + \alpha \times Loss_2$$

- **Evaluation Metrics:** Accuracy and F1-score









2 - MODEL DEVELOPMENT & TRAINING

Inference Configuration:

- **ONNX Conversion:** The best model was converted to ONNX format for optimized deployment and faster inference.
- **Model Size Optimization:** The model was resized and optimized, reducing its file size to 115 MB.
- **Deployment-Ready:** These adjustments ensure efficient resource usage and faster execution in production environments.
- **Visualizing Model Prediction (validation-set):**

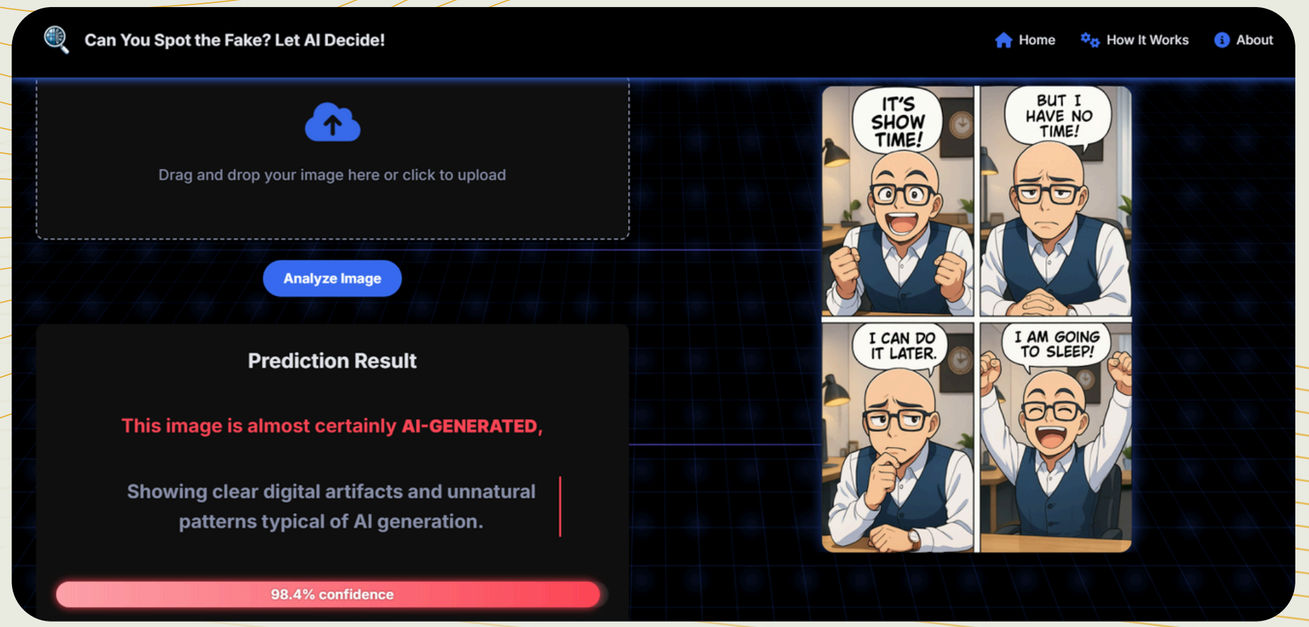
Our Kaggle competition score places us in the Top 75 teams from more than 550 teams, confirming the robustness of our approach and pipeline.

Private Score	Public Score
0.80988	0.77900

3- WEB BACK-END & DEPLOYMENT

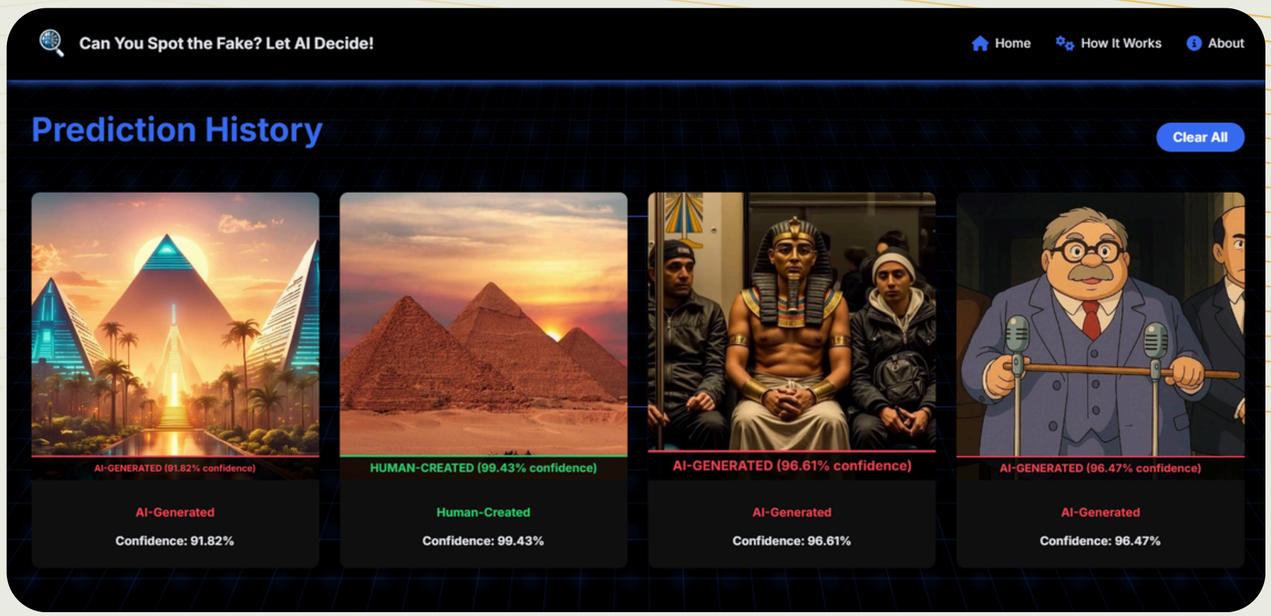
Frontend:

- Developed using HTML, CSS, and JavaScript, the interface is clean, interactive, and user-friendly.
- Users can download the prediction for their images with the predicted class and confidence printed on it for their records.
- It also offers a dedicated history section where users can view all their past predictions they saved.



Backend:

- Powered by FastAPI, the backend handles all image-upload routes and prediction requests efficiently.
- It performs all required image preprocessing and model inference, and includes comprehensive error handling to ensure reliable operation in production.
- Additionally, it integrates with other AWS services for model storage and containerized deployment.

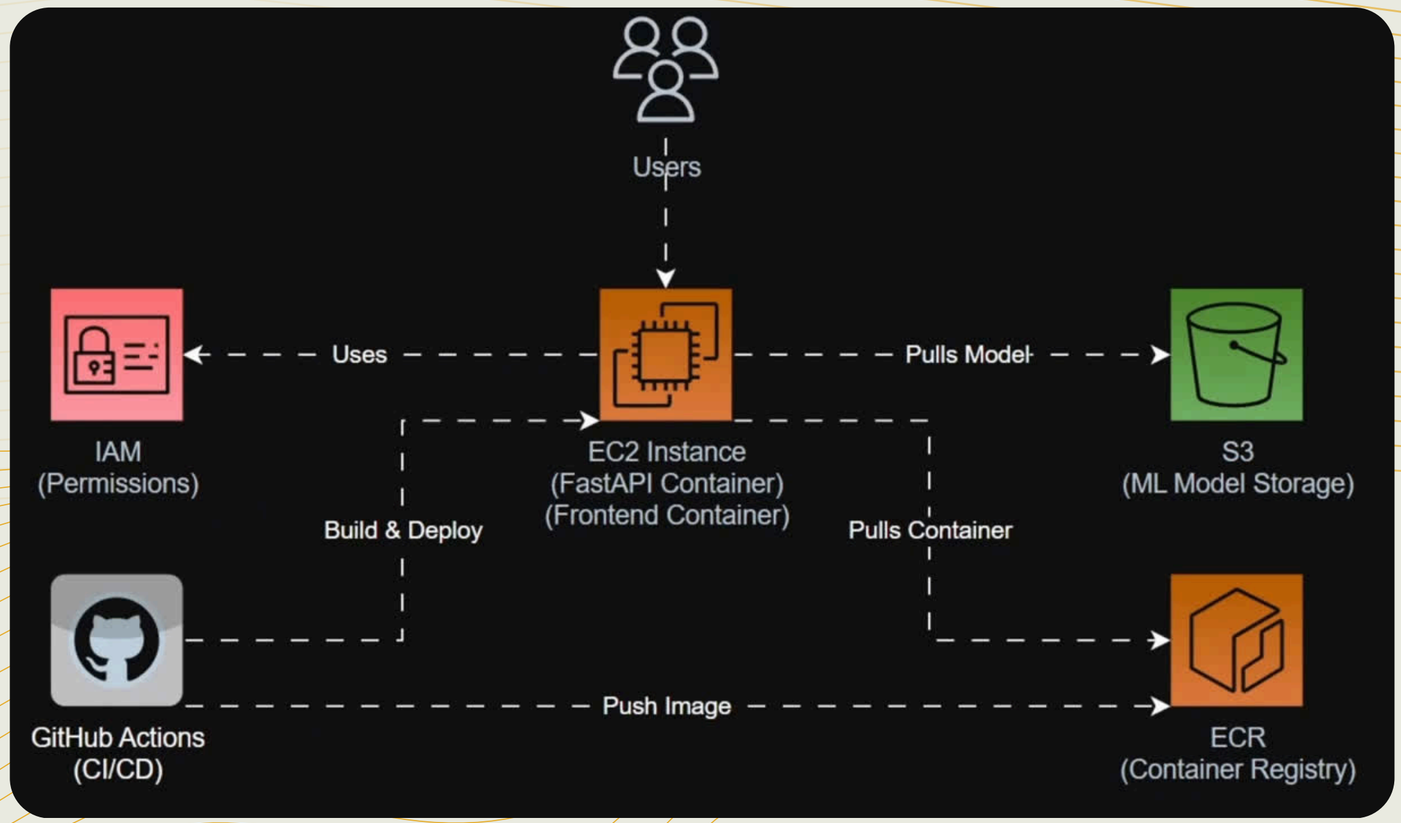


3- WEB BACKEND & DEPLOYMENT

Deployment Stack:

- Docker & Docker Compose: Used to containerize both the backend and frontend.
- AWS S3: Stores trained model files.
- AWS ECR: Hosts container images.
- AWS EC2: Pulls and runs the latest containers and models from ECR and S3.
- IAM Rules: Employed to securely grant EC2, least-privilege access to S3, ECR.
- GitHub Actions: Automates building and pushing of containers to ECR, triggering container updates on EC2.

System Architecture Design



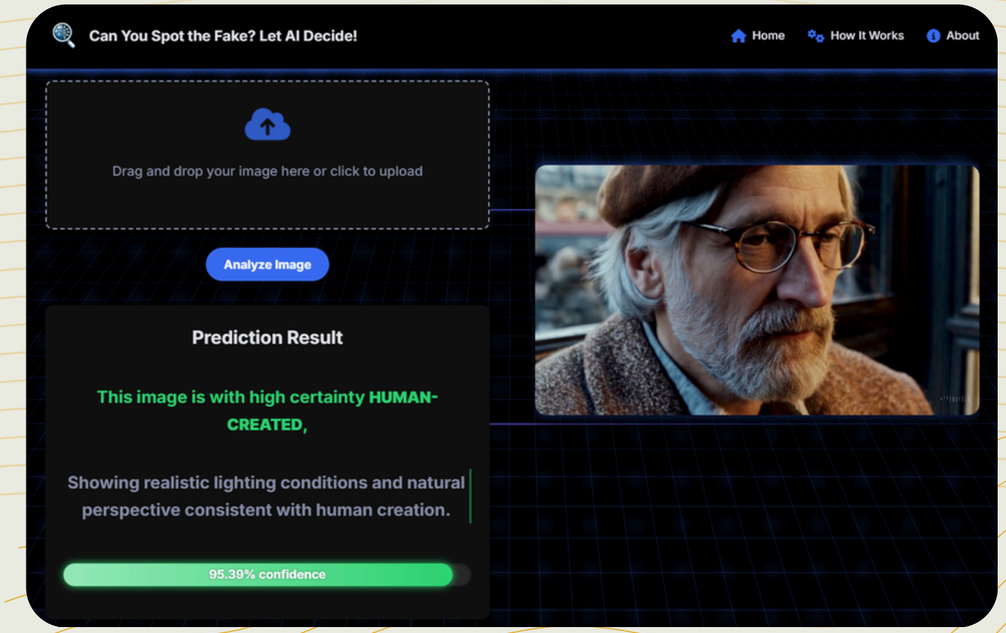
رواد مصر الرقمية

وزارة الاتصالات وتكنولوجيا المعلومات

FUTURE WORK & LIMITATIONS

Limitations

- **Data & Domain Drift:**
  - The model may underperform in production as generative models evolve rapidly, causing a drift between the training data and newly generated images.
- **Static Model:**
  - Without updates, the model risks becoming outdated, especially as image realism from AI tools continues to improve.
- **Lack of Interpretability:**
  - Currently, predictions are made as black-box outputs, with no real built-in explainability for users or developers.



Future Work

- **Continuous Retraining:**
  - Develop pipelines to frequently retrain the model with the latest AI-generated image data to keep pace with generative trends.
- **User Feedback Loop:**
  - Incorporate misclassified samples into the training set to enhance the system's robustness over time.
- **Explainable AI Integration:**
  - Add tools like Grad-CAM to help visualize model decisions and improve trust in predictions.

رواد مصر الرقمية

وزارة الاتصالات وتكنولوجيا المعلومات

CONCLUSION

We built a full-stack AI system to detect AI-generated images, from deep learning models to cloud deployment.

- Used advanced models (ConvNeXt, EfficientNet) with a custom fairness loss.
- Achieved Top 20 Scores on Kaggle (out of 550+ teams).
- Deployed a fast, lightweight ONNX model leveraging AWS Services along with Docker & Github Actions.
- Delivered a clean, interactive frontend with history, downloads, and error handling.

This project shows how to deploy scalable, fair, and real-world AI solutions to modern challenges.

The system is live and accessible at:

رواد مصر الرقمية

وزارة الاتصالات
وتكنولوجيا المعلومات

Thank you.

Questions..?

Team Members

Abdulrahman Khalid

Noureldin Youssef

Ahmed Mahmoud

Alaa Ashraf

Subervision By

Eng. Mohamed Qabalawy