PRAGATI ENGINEERING COLLEGE(AUTONOMOUS)

Approved by AICTE, permanently affiliated to JUNTUK Kakinada accredited by NBA & NAAC ‘A+’ Grade

Project Title:

CIFAKE: Image Classification and Explainable Identification of AI - Generated Synthetic Images

Project supervisor: Mrs. D. Chakra Satya Tulasi M.tech,

TEAM 08:

S. Annapurna (21A31A4427)
P. Samyuktha (21A31A4418)
B. Srikanth (21A31A4443)
G. Sneha Ratna (21A31A4410)
G. L. Shiva Teja (21A31A4445)

TABLE OF CONTENT

ABSTRACT

A synthetic dataset was generated using latent diffusion to create high-quality AI images that mimic the CIFAR-10 dataset, enabling comparison with real photographs. The project addressed a binary classification problem, distinguishing between real and AI-generated images, using Convolutional Neural Networks (CNNs) for classification. After training 36 CNN architectures and optimizing hyperparameters. Gradient Class Activation Mapping (Grad-CAM) was applied to interpret the model's decision-making process, revealing that the model relied on small background imperfections rather than the main entity in the image.

EXISTING SYSTEM OVERVIEW

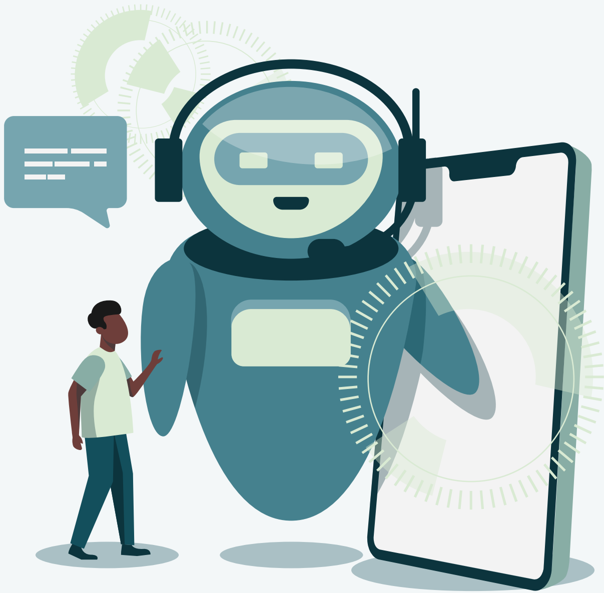
Existing system use advanced techniques like LDMs (e.g., Stable Diffusion, DALL-E, Imagen) to detect AI-generated images, which are visually complex and hard to identify. Detection methods such as DE-FAKE, EfficientNet, Vision Transformers, Optical Flow, and CNN-LSTM models show promise but still face challenges with high-quality images, generalization across datasets, and limited interpretability.

LIMITATIONS OF EXISTING SYSTEM

- **Reliance on Visual Glitches**: Methods depend on visible imperfections, which are rare in advanced models.
- **Lack of Interpretability**: Most models are black-boxes, making them unreliable for sensitive use cases.
- **Missed Subtle Features**: Current methods often overlook small imperfections in high-quality images.
- **Accuracy Challenges**: Existing methods struggle with low accuracy, especially for high-fidelity images.

PROPOSED SYSTEM OVERVIEW

The system employs a fine-tuned CNN to classify real and synthetic images, supported by Grad-CAM to visualize decision-making regions. It introduces the CIFAKE dataset with 60,000 real and 60,000 synthetic images. A dynamic retraining mechanism helps the model adapt to evolving synthetic image features. This approach focuses on detecting subtle visual anomalies. Overall, it offers improved performance compared to existing detection methods.



ADVANTAGES OF PROPOSED SYSTEM

- **Higher Accuracy**: The CNN achieves more accuracy, out performing existing methods for high-quality synthetic images.
- **Improved Explainability**: Grad-CAM visualizes key features, enhancing model transparency over black-box methods.
- **Dynamic Adaptability**: The system updates itself through retraining to handle new synthetic image challenges.
- **Focus on Subtle Imperfections**: The system detects small glitches and anomalies, improving detection of high-fidelity synthetic images.

SOFTWARE REQUIREMENTS

Operating System	Windows 10/11
Development Software	Python 3.10
Programming Language	Python
Integrated Development Environment (IDE)	Visual Studio Code
Front End Technologies	HTML5, CSS3, Java Script
Back End Technologies or Framework	Django
Database Language	SQL
Database (RDBMS)	MySQL
Database Software	WAMP or XAMPP Server
Web Server or Deployment Server	Django Application Development Server
Design/Modelling	Rational Rose

ALGORITHMS

01

CNN Algorithm

A deep learning model that learns features from images for tasks like classification.

02

Grad-CAM Algorithm

A technique that visualizes image areas influencing a model's prediction

CNN WORKFLOW

STEPS:-

1. Import necessary libraries
2. Loading the data
3. Data augmentation to improve generalization
4. Data Generators
5. CNN Model Architecture
6. Compile the model
7. Train the model
8. Save the model
9. Evaluate the model
10. Calculate accuracy

CODE PART:-

```
# CNN Model Architecture
model = Sequential([
    Conv2D(32, (3, 3), activation='relu',
input_shape=(IMG_WIDTH, IMG_HEIGHT, 3)),
    BatchNormalization(),
    MaxPooling2D(pool_size=(2, 2)),
    Dropout(0.25),
    Conv2D(64, (3, 3), activation='relu'),
    BatchNormalization(),
    MaxPooling2D(pool_size=(2, 2)),
    Dropout(0.25),
    Conv2D(128, (3, 3), activation='relu'),
    BatchNormalization(),
    MaxPooling2D(pool_size=(2, 2)),
    Dropout(0.25),
    Flatten(),
    Dense(128, activation='relu'),
    BatchNormalization(),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])
```

GRAD-CAM

Grad-CAM creates a heat map that highlights image regions most influential to a model's prediction. It uses feature maps from a convolutional layer and weights them using gradients of the predicted class. The resulting heat map reveals where the model is focusing, helping interpret deep learning decisions.

CODE PART:-

```
def compute_gradcam(img_array, model, layer_name='conv2d_3'):
    grad_model = tf.keras.models.Model([model.inputs], [model.get_layer(layer_name).output, model.output])
    with tf.GradientTape() as tape:
        conv_outputs, predictions = grad_model(img_array)
        loss = predictions[:, 0]
    grads = tape.gradient(loss, conv_outputs)
    guided_grads = tf.multiply(conv_outputs, grads)
    pooled_grads = tf.reduce_mean(guided_grads, axis=(0, 1, 2))
    conv_outputs = conv_outputs[0]
    heatmap = tf.reduce_sum(tf.multiply(pooled_grads, conv_outputs), axis=-1)
    heatmap = np.maximum(heatmap, 0) / tf.reduce_max(heatmap)  # Normalize the heatmap
    return heatmap.numpy()
```

SCREENSHOTS

HOME PAGE

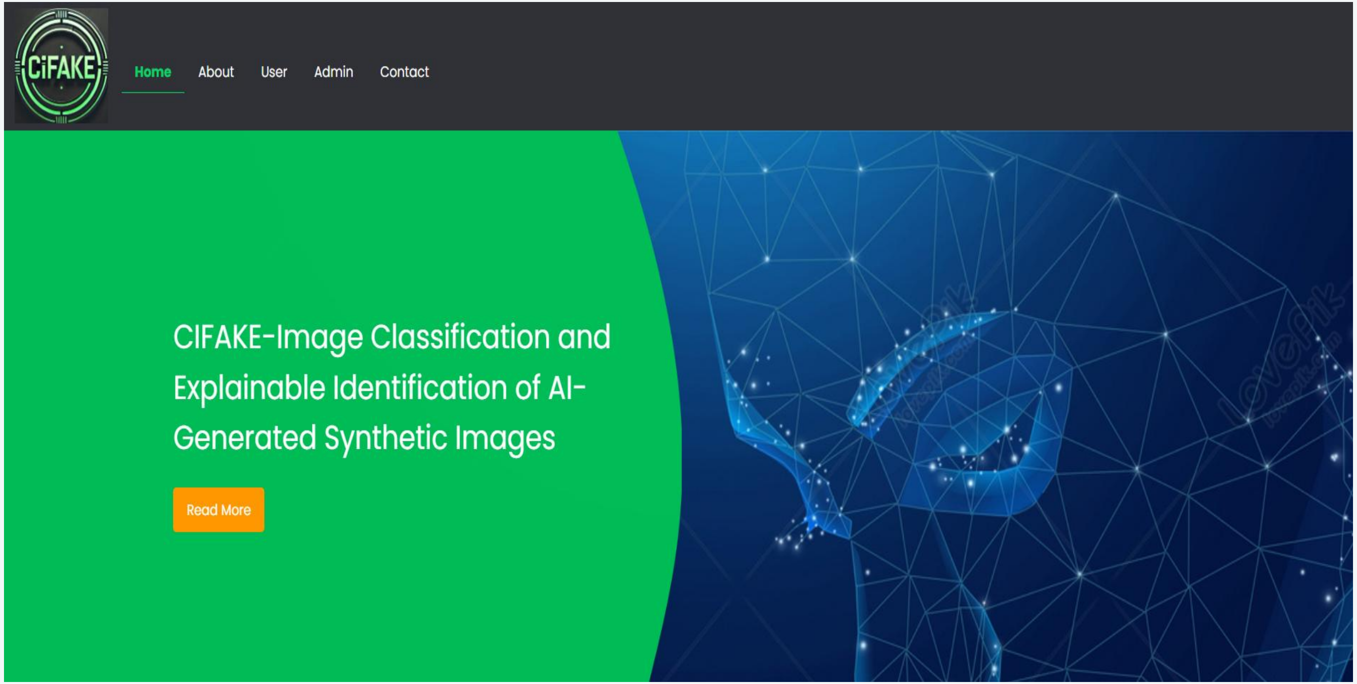


Fig-2 Home Page

ADMIN LOGIN PAGE

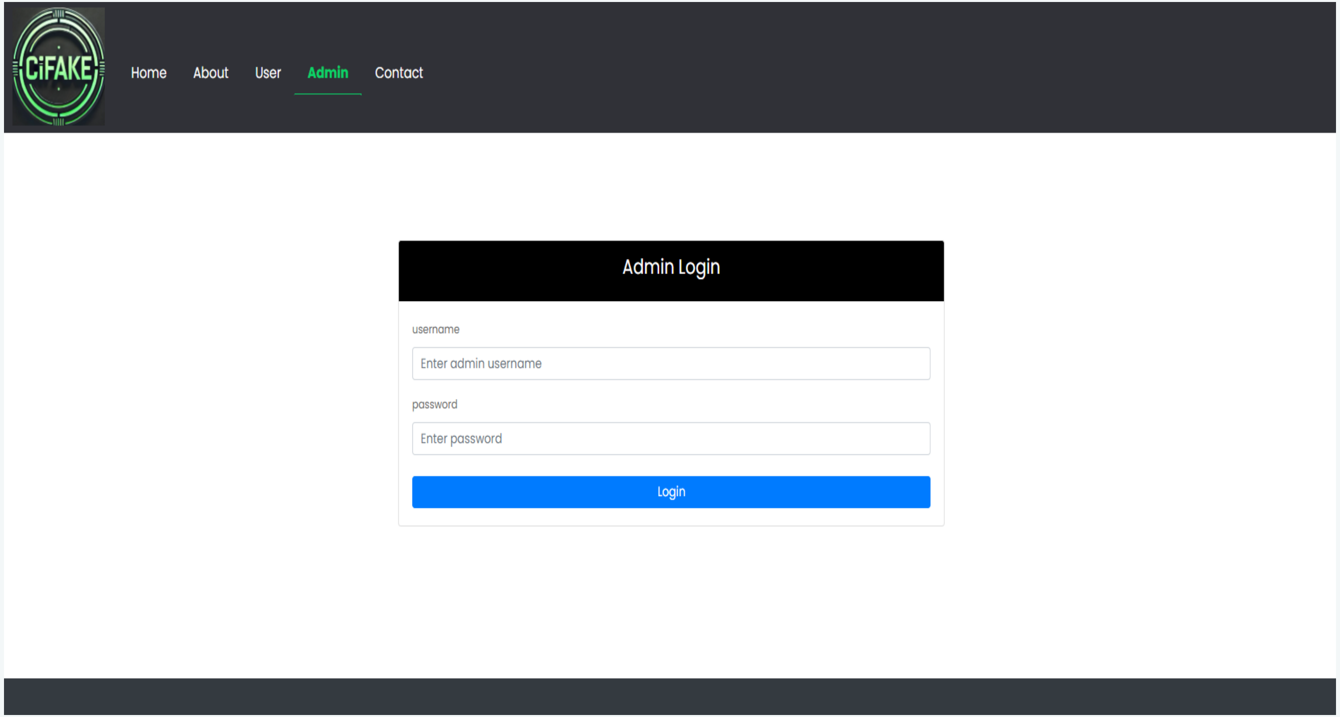


Fig-3 Admin Login Page

CNN MODEL

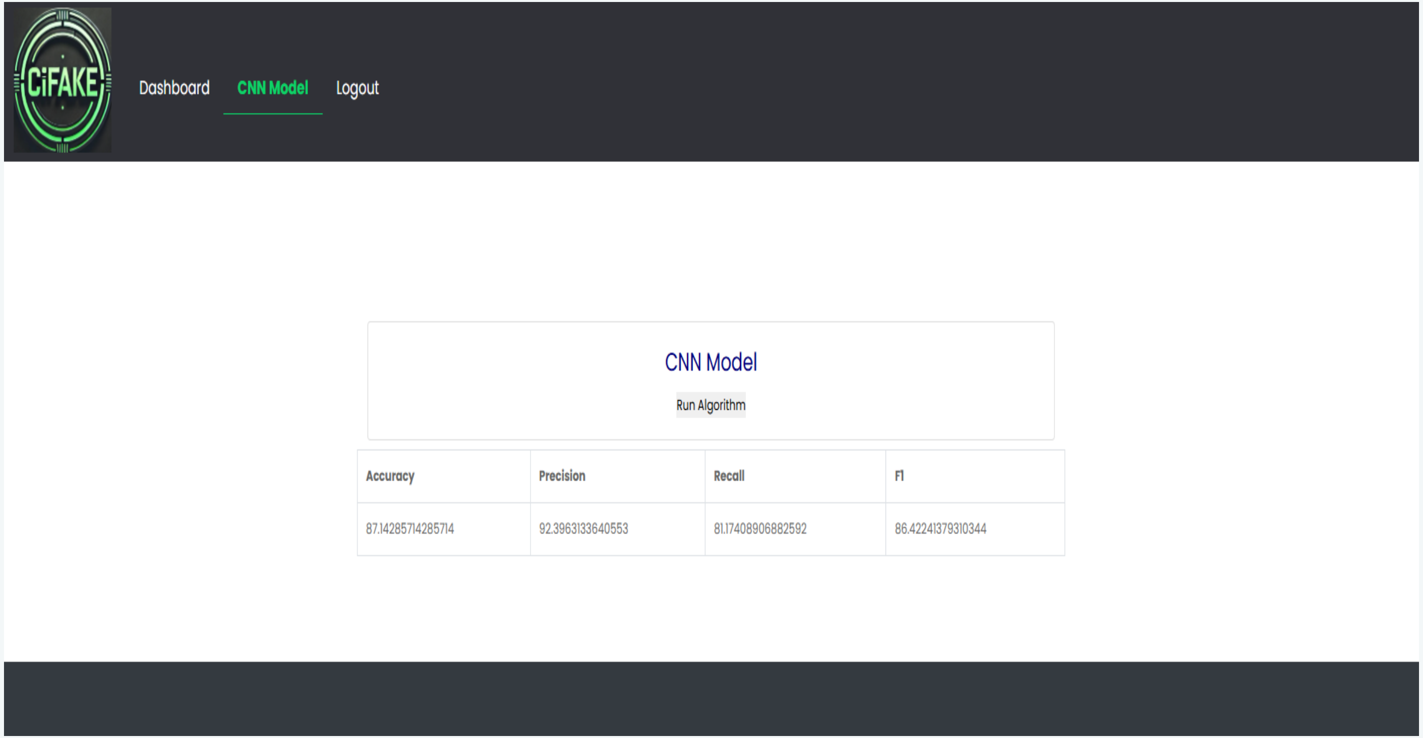


Fig-4 CNN Model Page

USER LOGIN AND REGISTRATION PAGE

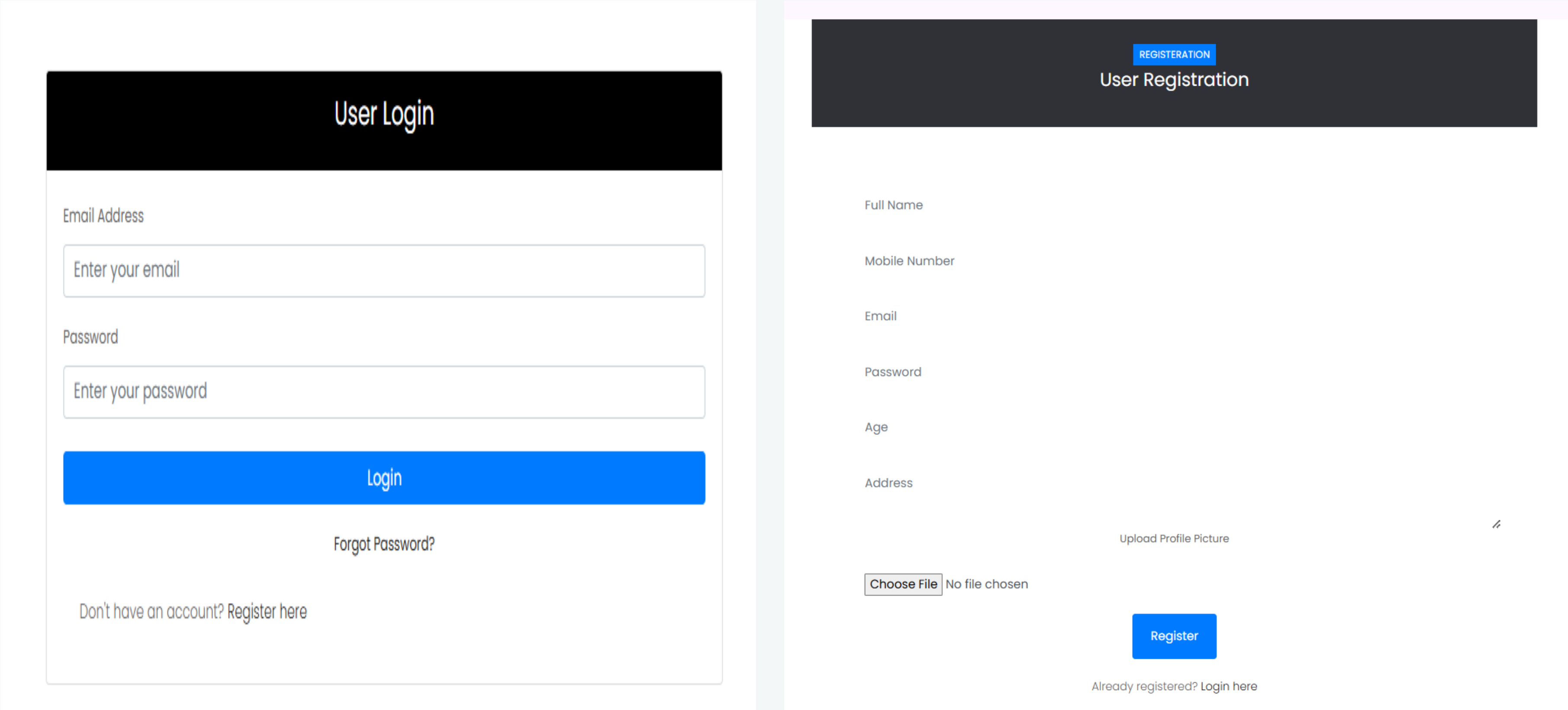


Fig-5 User Login and Registration Page

PREDICTION PAGE

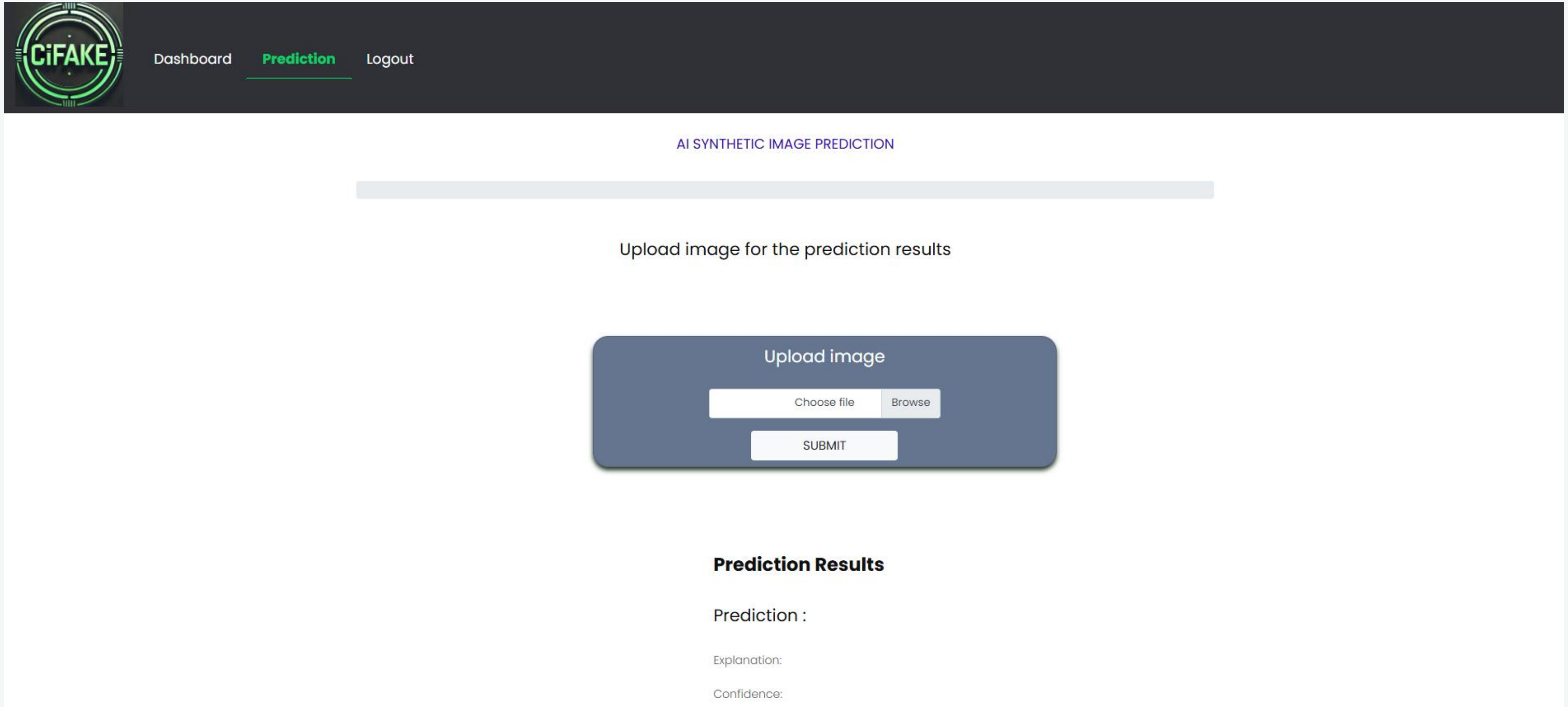


Fig-6 Prediction Page

PREDICTION RESULTS (REAL OR FAKE)

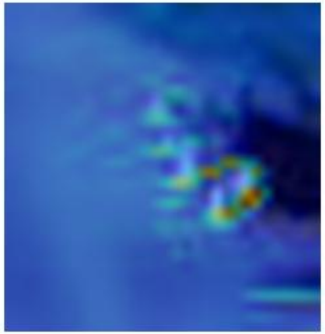
Prediction Results

Prediction : Fake

Explanation: The image contains patterns that resemble known fake characteristics.

Confidence: 92.31%

Grad-CAM Visualization



Prediction Results

Prediction : Real

Explanation: The image shows features highly similar to real images.

Confidence: 94.49%

Grad-CAM Visualization

Fig-7 Prediction Results(Real or Fake)

FUTURE IMPLEMENTATION

Future work can focus on attention-based models like Transformers to boost prediction accuracy and interpretability. Updating the CIFAKE dataset and expanding it to other domains, such as healthcare or facial recognition, could enhance its applicability. Incorporating Explainable AI (XAI) techniques will further improve transparency and help build trust in predictions, supporting better decision-making.

CONCLUSION

The project significantly improved the accuracy of recognizing AI-generated images and introduced the CIFAKE dataset to support future research. It addresses the critical challenge of verifying the authenticity of visual data, which is essential in combating misinformation and protecting digital integrity. By combining advanced deep learning techniques with explainable AI, the system enhances trust and transparency. The dataset also provides a valuable resource for benchmarking and developing robust detection models across multiple domains.

THANK YOU

Any Questions?