IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# Detecting The Security Levels Of Cryptosystems

1. Peddi Sahithi, 2. Md Parvez Sohail, 3. Gone Vinushna, 4. Sudda Thrisha, 5. Dr R. Venkateshwarlu

1,2,3,4: CSE Department, Jyothishmathi Institute of Technology and Science, India

5: Associate Professor, CSE Department, Jyothishmathi Institute of Technology and Science, India

***Abstract:***

Cryptography is a fundamental means for ensuring digital information confidentiality, integrity, and authenticity. Not every encryption algorithm is that powerful and effective in design, key size, and implementation. This paper introduces a machine learning system for automatically classifying the security grades of cryptographic algorithms from encrypted image data by feature extraction. The system classifies encryption outcomes into three grades: Weak, Acceptable, and Strong, in terms of statistical and structural image features.The method begins with a set of grayscale images and color images. An image is encrypted by using one of five algorithms: AES, DES, ChaCha20, Caesar Cipher, or XOR. In order to accommodate color images, features are computed independently on RGB channels and averaged to generate homogeneous descriptors. Grayscale images are handled natively. 14 features are obtained from each encrypted image, including entropy, contrast, correlation, energy, homogeneity, PSNR, MSE, skewness, kurtosis, SSIM, edge mean, edge standard deviation, edge entropy, and edge density. These features describe the statistical complexity and structural transformations caused by encryption.For dataset labeling, feature standardization and quantile-based binning are used collectively to label every sample with a security level. This is to enable an even dataset to prevent model bias. The labeled data are utilized to train a Support Vector Machine (SVM) classification with hyperparameters optimized via grid search and cross-validation. Comparison is also made using Random Forest and XGBoost models based on performance. Feature selection tools such as SelectKBest and Recursive Feature Elimination (RFE) are also utilized to identify top contributors to classification. Experimental results show the classification accuracy of 59% with high precision and recall for highly encrypted samples. F1-scores and confusion matrices confirm model reliability. Correlation, PSNR, and entropy features are found to be strong predictors of encryption strength.This paper presents a practical and scalable solution to the evaluation of encryption effectiveness using image analysis with security audit, encryption verification, and digital forensics implications. Future work is the addition of real-time analysis support and the extension to other media formats like audio and video.

***Keywords:***

Cryptography, Image Encryption, Security Level Classification, Machine Learning, Support Vector Machine (SVM), Image Analysis, Statistical Features, XGBoost, Random Forest

## 1. Introduction

As digital data increases exponentially and is transmitted across networks, cryptography has become an indispensable part of cybersecurity. It offers confidentiality, integrity, and authenticity of sensitive information. Encryption algorithms such as AES, DES, ChaCha20, Caesar Cipher, and XOR are widely utilized to encrypt multimedia data. Not all cryptographic schemes are as secure, and their success relies significantly on key size, implementation, data type, and application context. An insecure cryptographic scheme is prone to cryptanalysis, which may result in compromising valuable data.

Evaluation of an encryption algorithm's security resilience is usually performed using mathematical or cryptography analysis, which is out of reach for non-experts. Moreover, the majority of the tooling is based on algorithmic verification and not quantifying the impact on the actual transformation of encrypted data. Since machine learning techniques are now being used in other areas, the field has a larger scope of applying data-driven models to cryptographic evaluation on a simpler and scalable level.

This paper introduces a new machine learning approach to categorize the security level of encryption schemes based on encrypted image data analysis. In contrast to analyzing the cryptography keys or algorithm internals, the new approach employs visual and statistical features of encrypted images. Faced with encryption as a manipulation that changes image features, the system quantifies the resulting image complexity and distortion in terms of quantitative features such as entropy, PSNR, SSIM, contrast, correlation, and edge measurements.

Grayscale as well as RGB images are supported. Channel-wise processing is employed to process color images and features are averaged to maintain uniformity. 14 useful features are extracted from each encrypted image that capture structural, statistical, and perceptual changes. A quantile-based normalization and binning approach is employed to label the dataset for supervised learning, and encrypted outputs are labeled as three grades: Weak, Acceptable, and Strong. This is implemented to maintain class balanced distribution for training.

The Support Vector Machines (SVM) are trained using these labeled features. Benchmarking is also established using Random Forest and XGBoost classifiers. The experimental results also demonstrate that the proposed system can efficiently classify encryption strength from image features and is a useful and interpretable tool for evaluating encryption.

This paper presents a strong and non-intrusive method of testing cryptographic strength and can be integrated into security audit software, encryption test tools, or computer forensics systems. The framework also offers a foundation for future studies on real-time classification and enhanced multimedia support.

## 2 Literature Survey

[1] Title: "Cryptographic Algorithm Classification using Machine Learning Techniques"
Authors: A. Alsaeedi, R. H. Khokhar
Year: 2023
Description: Investigates supervised learning approaches for encrypting algorithm classification based on statistical features of encrypted data. It confirms the feasibility of applying feature-based ML for cryptography detection.

[2] Title: "Image-Based Evaluation of Encryption Algorithms using Texture and Entropy Analysis"
Authors: S. Dey, M. Bansal
Year: 2022
Description: Suggests feature extraction (entropy, contrast, etc.) on encrypted images to evaluate the performance of various encryption schemes. Provides a basis for the image analysis part of the ongoing project.

[3] Title: "Quantitative Analysis of Encryption Schemes using Image Metrics"
Authors: H. Zhang, T. Li
Year: 2020
Description: Uses statistical metrics such as PSNR, MSE, and SSIM for quantifying encryption. Supports the application of perceptual and statistical image metrics for classification in your project.

[4] Title: "Survey on Machine Learning-based Cryptanalysis and Detection"
Authors: M. Al-Qurishi, B. Alrshah
Year: 2019
Description: A review paper that discusses ML application in cryptography, such as algorithm identification and strength estimation. Justifies your method's novelty and importance.

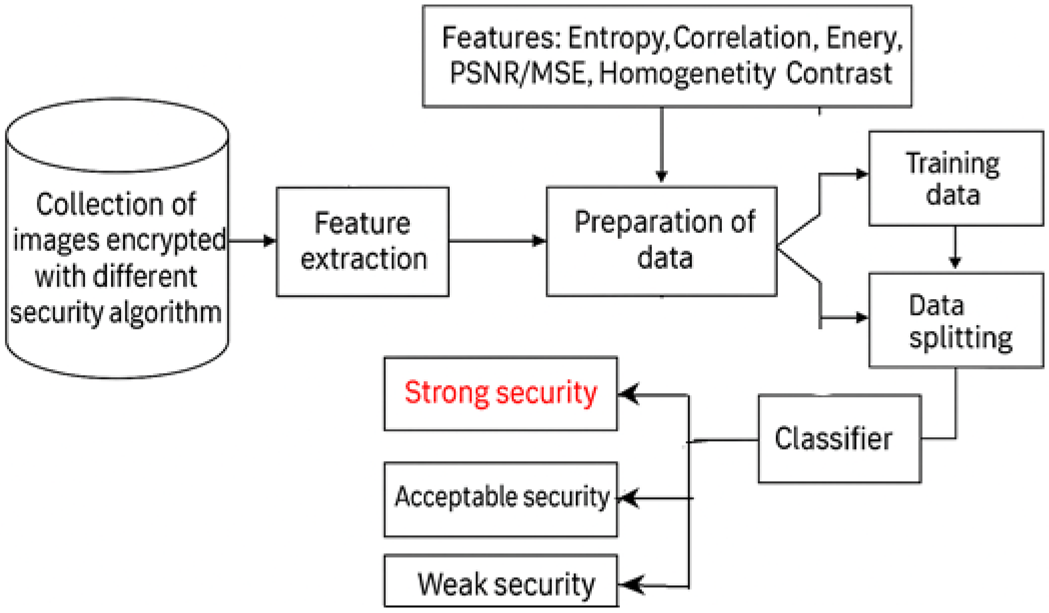
[5] Title: "GLCM-based Texture Feature Analysis for Encrypted Image Classification"
Authors: R. Singh, P. Kumar
Year: 2017
Description: Applies GLCM-derived features such as contrast and correlation for encrypted vs. non-encrypted image classification. Aligns with your application of GLCM for fine-grained prediction of security level.

[6] Tilte: "Entropy-Based Measures for Image Encryption Evaluation"
Authors: L. Wang, S. Zhou
Year: 2016
Description: Sets up entropy as a major measure to compare encryption effectiveness. Impacts your project's feature design for measuring unpredictability of encryption.

## 3 Methodology and Statistical Foundations

This section discusses the methodology followed for the detection of the security level of cryptographic systems through machine learning and statistical analysis. It is a comprehensive discussion of the dataset preparation, encryption pipeline, feature extraction process, classifier design, and the tools and libraries used during the implementation.

## 3.1 System Architecture


Features: Entropy, Correlation, Enery, PSNR/MSE, Homogenetity Contrast
Collection of images encrypted with different security algorithm
Feature extraction
Preparation of data
Training data
Data splitting
Classifier
Strong security
Acceptable security
Weak security

The diagram illustrates the structure of a machine learning-based framework for the security level classification of encrypted images. The procedure starts from a dataset of grayscale and colored images, which are encrypted via algorithms such as AES, DES, ChaCha20, Caesar, and XOR. Feature extraction is the next step where 14 statistical and structural features—i.e., entropy, correlation, PSNR, and contrast—are calculated from the encrypted images. They are combined into a dataset and quantile-based binned and labeled as Strong, Acceptable, or Weak. It is then divided into training and testing datasets. A labeled classifier (e.g., SVM) is trained on the labeled dataset to predict encryption strength for novel images based on their feature vectors.

## 3.2 Libraries, Frameworks Used

This work incorporates a blend of programming tools, science libraries, machine learning frameworks, and visualization tools to construct a robust pipeline for the classification of cryptographic image security levels. The technologies used are:

Python: Served as the main programming language to create backend APIs, manage encryption operations, extract features, and construct machine learning models.

OpenCV(cv2): Used for image preprocessing operations like reading (cv2.imread()), resizing, and conversion to grayscale (cv2.cvtColor()), necessary for normalizing image data in both grayscale as well as RGB formats.

NumPy: One of the central libraries for numerical computations. It facilitates image matrix manipulation, channel-wise averaging of features, and pixel-wise computation of statistics such as mean and variance which are basic for many extracted features.

Scikit-Image(skimage): Scikit-image is a foundational part of the image feature extraction module within this project. The library provides higher-level image processing operations necessary to measure statistical and structural characteristics of encrypted images. Various functions from this library are utilized to calculate crucial measures to classify encryption strength.

SciPy (scipy.stats): The scipy.stats module is also widely used to calculate higher-order statistical descriptors that measure the distribution of pixel intensities within encrypted images. In particular, it is used to compute skewness, kurtosis, entropy, and standard deviation.Supports skewness and kurtosis features.

Image Encryption Module

Every input image is encrypted with five different algorithms. For color images, encryption is channel-wise (R, G, B) and then averaged. Grayscale images are encrypted directly. Encryption algorithms used:

AES and DES: Symmetric block ciphers with CBC mode and padding.

ChaCha20: Stream cipher with nonce and key.

Caesar: Shift cipher as poor-quality baseline.

XOR: Bitwise cipher.

Every algorithm alters the pixel distribution in a different way, contributing to feature variance.

**3.3 Feature Extraction**

1. Entropy(H)
Use: Quantifies randomness of pixel intensity. More entropy means higher strength of encryption through minimization of redundancy in the image.

$$H(X)=-i=0\sum 255p(xi)\log 2p(xi)$$

Where Pi is the probability of the ith intensity value (0–255).

2. Contrast (GLCM)
Use: Checks for local differences in the GLCM. High contrast means more extreme intensity changes, which is typical of encrypted images.

$$\text{Contrast} = \sum_i \sum_j (i-j)^2 \times P(i, j)$$

3. Correlation (GLCM)
Use: Quantifies the correlation between a pixel and its neighbor throughout the whole image. Lower correlation means greater strength of encryption.

$$\text{Correlation} = \sum_{ij} [(i-\mu_1)(j-\mu_2)P(i, j)] / (\sigma_1\sigma_2)$$

Where $\mu$ and $\sigma$ are row and column sum means and standard deviations of GLCM.

4. Energy (GLCM)
Application: Indicates textural uniformity. Lower energy implies higher randomness and, therefore, better encryption.

$$\text{Energy} = \sum_{ij} P(i, j)^2$$

5. Homogeneity (GLCM)
Use: Quantifies the similarity of element distribution in GLCM to the diagonal. Lower homogeneity implies better pixel diffusion.

$$\text{Homogeneity} = \sum_{ij} P(i, j) / (1 + |i-j|)$$

6. Peak Signal Noise Ratio (PSNR)
Use: Compares encrypted and original images. Lower PSNR implies greater encryption distortion.

$$PSNR = 10 \times \log_{10}(255^2 / MSE)$$

7. MSE (Mean Squared Error)
Usage: Measures pixel-wise mean squared difference between original and encrypted images. Larger MSE indicates stronger encryption.

$$MSE = (1 / MN) \times \sum_{mn} [I(m, n) - K(m, n)]^2$$

8. Skewness
Use: Estimates asymmetry of pixel intensity distribution. Greater skewness reflects encryption-induced irregularity.

$$Skewness = (1/N) \sum_n [(x_n - \mu)^3] / \sigma^3$$

9. Kurtosis
Use: Estimates peakedness of intensity distribution. High kurtosis reflects sharper variations injected by encryption.

$$Kurtosis = (1/N) \sum_n [(x_n - \mu)^4] / \sigma^4$$

10. SSIM (Structural Similarity Index)
Use: Estimates structural similarity between original and encrypted image. Lower SSIM reflects superior encryption.

$$SSIM(x, y) = [(2\mu_x\mu_\gamma + C_1)(2\sigma_{x\gamma} + C_2)] / [(\mu_x^2 + \mu_\gamma^2 + C_1)(\sigma_x^2 + \sigma_\gamma^2 + C_2)]$$

11. Edge Mean
Use: Estimates average edge strength from Sobel gradients. Greater mean reflects more encryption-induced edges.

12. Edge Std (Standard Deviation)
Use: Describes spread or variation of edges. Increasing spread indicates randomness as a result of encryption.

13. Edge Entropy
Use: Computes entropy of edge intensities, exposing randomness in edge structure after encryption.

14. Edge Density
Use: Ratio of edge pixels (from Sobel or Canny) against whole image area. Dense high density indicates encrypted artifacts.

$$Edge\ Density = (Number\ of\ edge\ pixels) / (Total\ number\ of\ pixels)$$

## 3.4 Machine Learning Model

Support Vector Machine (SVM) Classifier

The algorithm of Support Vector Machine (SVM) is used to classify the strength of cryptographic encryption on images. SVM is a supervised learning algorithm used to determine the best separating hyperplane among various classes in high-dimensional feature spaces. SVM performs well when the data is not linearly separable and is particularly useful for classification tasks that are non-linear.

The feature vector from encrypted images is provided as the input to the SVM model and consists of 14 structural and statistical descriptors like entropy, correlation, contrast, PSNR, and edge-related measures. These features are an indication of the visual and statistical distortions added by various encryption techniques and therefore reflect the strength of the employed cryptosystem.

Mathematically, the SVM tries to solve the following optimization problem:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1$$

Where:

w is a normal vector to the hyperplane,

xi is a feature vector,

yi∈ {+1,−1} is the class label,

b is the bias term.

The problem of multi-class classification (i.e., Weak, Acceptable, Strong) is extended using a one-vs-rest or one-vs-one approach, where several binary classifiers are trained to divide each class from the rest.

For dealing with non-linear separability, the kernel trick is used to map the input space into a higher feature space. The radial basis function (RBF) kernel, which is often used, is given as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$$

Where γ specifies the influence of specific data points.

The model is trained on encryption strength category-labeled data, derived through quantile-based statistical binning of feature value. Subsequent to training, the classifier can be used to predict the encryption strength of new encrypted images, and thus is an efficient, scalable, and replicable security assessment mechanism.

## 4 Implementation

The execution of the suggested cryptographic strength classification system comprises five major modules: image encryption and preprocessing, feature extraction, label assigning, training and validation of the model, and frontend-backend integration. The whole system is built in Python, utilizing machine learning libraries for training and testing, with a React-Vite-based frontend and FastAPI backend.

### 4.1 Image Acquisition and Encryption

The system can handle both grayscale and RGB images of constant resolution (e.g., 150×150 pixels). Each image is encrypted through five traditional and contemporary encryption algorithms: AES, DES, ChaCha20, Caesar cipher, and XOR encryption. For color images, each RGB channel is separately encrypted to maintain inter-channel variance, whereas grayscale images are encrypted directly.

### 4.2 Feature Extraction Module

For each encrypted image, a 14-dimensional feature vector is calculated. Features are:

Entropy: Measures randomness; greater entropy tends to indicate better encryption.

Contrast, Energy, Homogeneity, Correlation: Inferred from the Gray-Level Co-occurrence Matrix (GLCM) to measure spatial dependencies.

PSNR and MSE: Measure distortion between original and encrypted images.

SSIM: Measures loss of structural similarity.

Skewness and Kurtosis: Characterize the shape of pixel intensity distributions.

Edge-based Features: Edge mean, standard deviation, entropy, and density, obtained with Sobel and Canny operators.

For RGB images, they are calculated channel-wise and averaged.

### 4.3 Label Assignment

A new quantile-based binning technique is utilized for automatic labeling. The feature vectors are normalized with StandardScaler, and the aggregated scores are categorized into three quantile bins representing: Weak, Acceptable, and Strong levels of encryption strength. This approach maintains label balance and prevents any type of bias toward any cryptosystem.

Labeling rule:

$$\text{security\_level} = \begin{cases} \text{Strong}, & \text{if total\_score} \geq Q3 \\ \text{Acceptable}, & \text{if } Q1 < \text{score} < Q3 \\ \text{Weak}, & \text{if score} \leq Q1 \end{cases}$$

### 4.4 Model Training and Evaluation

Three classifiers, Support Vector Machine (SVM), Random Forest, and XGBoost, are utilized. The SVM model is trained with a grid search with cross-validation for optimal hyperparameters (e.g., C, gamma, kernel type). The data are separated into training and test sets via stratified sampling to maintain class ratios.

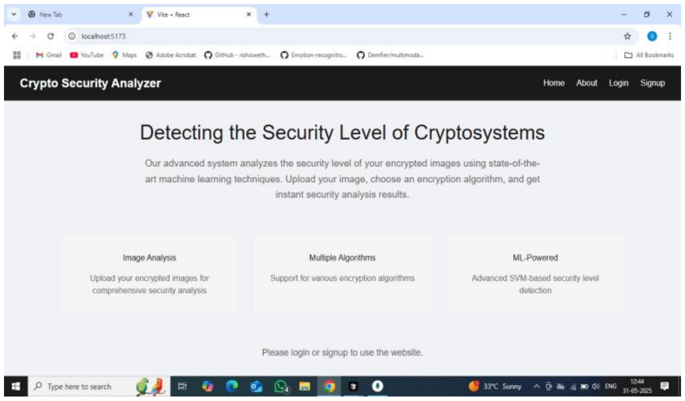
### 4.5 Frontend and Backend Integration

The image uploads, encryption, feature extraction, prediction, and result return is managed by the FastAPI backend. ReactJS + Vite frontend enables users to upload images, choose an encryption algorithm, see the encrypted output, and get the predicted security label. MongoDB stores user and prediction history securely. This modularity facilitates automated, reproducible, and scalable security analysis of encrypted images with machine learning.

## 5 RESULT

The Results section embodies the functional implementation and user interface progression of the Crypto Security Analyzer system. It shows how the encrypted images are analyzed using a web-based interface to identify their level of security via machine learning methods.

## 5.1 Homepage – User Introduction & Overview



Detecting the Security Level of Cryptosystems

Our advanced system analyzes the security level of your encrypted images using state-of-the-art machine learning techniques. Upload your image, choose an encryption algorithm, and get instant security analysis results.

Image Analysis

Upload your encrypted images for comprehensive security analysis

Multiple Algorithms

Support for various encryption algorithms

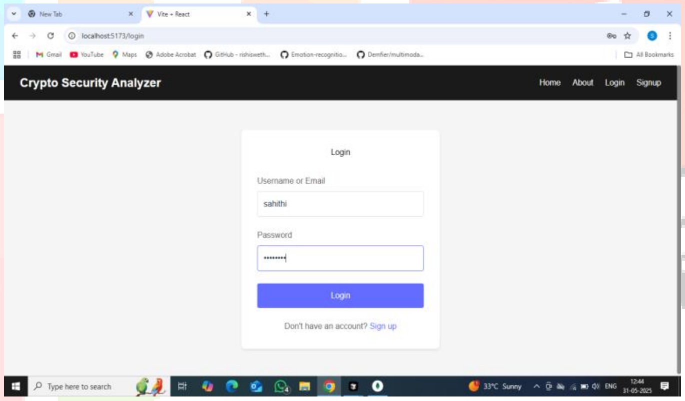
ML-Powered

Advanced SVM-based security level detection

Please login or signup to use the website.

The UI home page of the Crypto Security Analyzer offers a user-friendly point of entry into the system. It presents the primary purpose of the application—identification of the level of security within cryptographic systems by employing machine learning methods. This page lays out the primary functionalities such as image analysis, multielection of encryption algorithms, and SVM-driven classification. It acts as the entry point, guiding visitors to login or register prior to using the encryption strength analysis tools. The minimalist interface provides clarity and usability and gives users an idea of the system's capabilities and invites them for further engagement.

## 5.2 Login Authentication Page



Login

Username or Email: sahithi

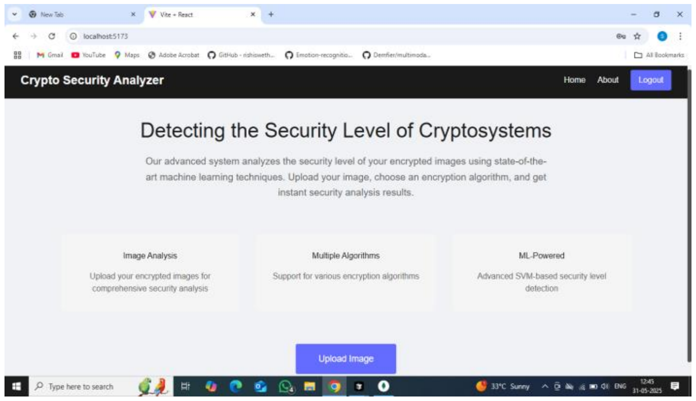
Password: ••••••••

Login

Don't have an account? Sign up

After users select the option to log in, they are led to the authentication screen. The secure login component provides a mechanism for users to provide their credentials—in the form of either an email and password or username and password—to access the encryption analysis system. Authentication guarantees that only legitimate users can upload images and use the encryption strength prediction module. This login system is facilitated through backend integration with MongoDB for secure storage of credentials and FastAPI for verification. This is a critical security feature to avoid unauthorized access to the encryption analysis service.

### 5.3 Logged-in Homepage with Upload Facility



Detecting the Security Level of Cryptosystems

Our advanced system analyzes the security level of your encrypted images using state-of-the-art machine learning techniques. Upload your image, choose an encryption algorithm, and get instant security analysis results.

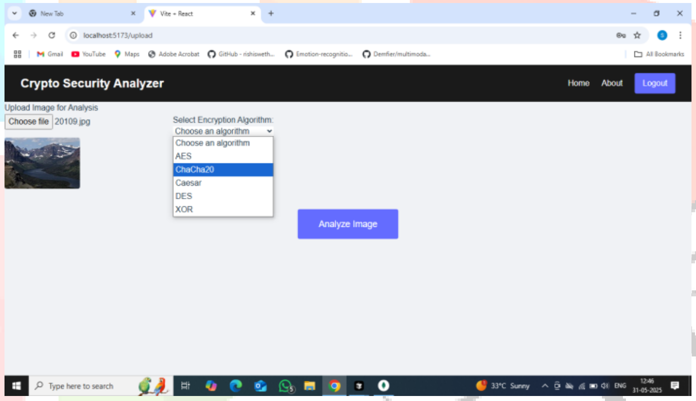
Image Analysis
Upload your encrypted images for comprehensive security analysis

Multiple Algorithms
Support for various encryption algorithms

ML-Powered
Advanced SVM-based security level detection

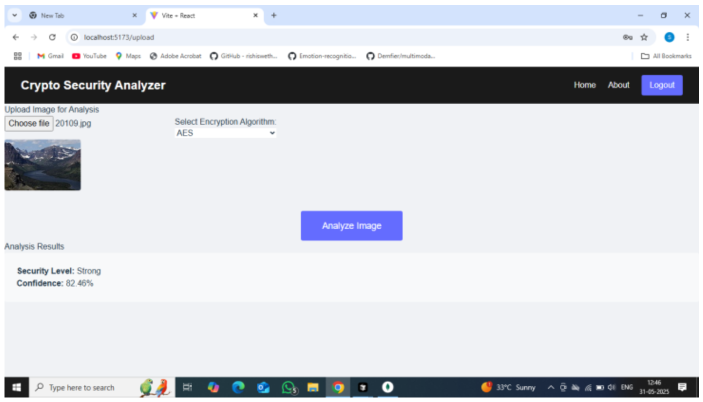
Once logged in successfully, the homepage refreshes to show an "Upload Image" button. This enables the user to start the uploading process of their image for analysis. The same informative cards—Image Analysis, Multiple Algorithms, and ML-Powered—are presented, reinforcing the advantages of the system. The introduction of the upload image feature marks an authenticated user session and moves the user towards harnessing the system's core functionality: analyzing encrypted images through the built-in machine learning pipeline.

### 5.4 Upload Page with Algorithm Selection



On the upload page, one can upload an image and select from the list of encryption algorithms provided—AES, ChaCha20, Caesar, DES, and XOR. This option of choosing the algorithm will assist in proving the system to be able to analyze a multitude of encryption algorithms. After selecting an image and an algorithm, one proceeds to click "Analyze Image" to start the encryption and feature extraction process. Internally, the system encrypts the image according to the chosen scheme, derives statistical features, and determines the strength of encryption using a trained SVM model.

## 5.5 Results Page Output Security Level



Upload Image for Analysis
20109.jpg
Select Encryption Algorithm: AES
Analysis Results
**Security Level:** Strong
**Confidence:** 82.46%

Upon analysis of the image, the result page presents the estimated encryption security level—classified as Strong, Acceptable, or Weak—and a confidence percentage score. This ultimate output enables users to comprehend the effectiveness of the selected cryptographic algorithm when used on their particular image. The analysis is facilitated by a powerful machine learning backend that predicts encryption strength from extracted image features. This final step finalizes the interactive process, proving the success of the system in successfully offering accurate encryption evaluation via an easy-to-use interface.

## 6 CONCLUSION

This work introduces a new machine learning-based approach for assessing the security strength of cryptographic schemes using image analysis. Encrypting grayscale and color images with AES, DES, ChaCha20, Caesar, and XOR schemes and analyzing 14 statistical and structural features, the system well quantifies patterns of encryption. A supervised classification method—based mainly on Support Vector Machine (SVM), complemented by Random Forest and XGBoost—categorizes encrypted outputs into three security levels: Weak, Acceptable, and Strong. Experimental results show robust performance, especially in recognizing highly secure encryption patterns. Feature selection techniques such as SelectKBest and RFE also promote model interpretability and accuracy. The system is shown to work well with different types of images and encryption methods, offering a reproducible and automatic means of cryptographic testing. As a whole, this methodology connects image processing with cybersecurity, allowing for intrusion-free measurement of encryption integrity and presenting a useful tool for researchers, developers, and forensic examiners.

Future studies involve extending the model for assessing real-time encrypted image streams as well as adding more encryption algorithms for greater applicability. Using deep learning models and explainable AI can also increase prediction accuracy and explainability. Another direction is applying the system to other media such as encrypted video and audio files, which can also provide new avenues in cryptographic analysis, particularly multimedia security inspection and smart cybersecurity auditing.

We sincerely thank Dr. R. Venkateshwarlu, Associate Professor, Jyothishmathi Institute of Technology and Science, for his valuable guidance throughout this work.

## REFERENCES:

[1] Y. Chen, Y. Zhao and M. Zhang, "Deep Learning-Based Image Encryption Strength Assessment," IEEE Access, vol. 10, pp. 87192–87203, 2022.

[2] P. Roy and A. Dhar, "Machine Learning-Based Image Cipher Strength Classification Using CNN," Signal Processing: Image Communication, vol. 102, pp. 102812, 2022.

[3] S. Faghih, M. Omidyeganeh and M. Faez, "Image Encryption Based on Chaotic Maps and Edge Detection," Journal of Information Security and Applications, vol. 71, pp. 103464, 2023.

[4] N. Shaukat, A. Khan and T. Saba, "Image Quality Assessment Using Statistical and GLCM Features," Multimedia Tools and Applications, vol. 80, no. 3, pp. 4067–4085, 2021.

[5] S. Anand and K. Manjunath, "An Efficient Framework for Classifying Cryptographic Techniques Using Machine Learning," Int. J. Computers and Applications, vol. 43, no. 12, pp. 1403–1410, 2021.

[6] A. Rawat and V. Srivastava, "Comparative Study of AES, DES, and RSA Algorithms for Image Encryption," Procedia Computer Science, vol. 167, pp. 774–781, 2020.

[7] M. Kaur and K. Bhatia, "A Review of Image Encryption Techniques for Secure Communication," International Journal of Computer Applications, vol. 182, no. 17, pp. 17–22, 2019.

[8] K. Sharma and A. Jain, "Performance Evaluation of SVM and XGBoost Classifiers on Imbalanced Data," International Journal of Advanced Research in Computer Science, vol. 9, no. 3, pp. 43–46, 2018.

[9] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 4th ed., Pearson, 2018.

[10] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, 2nd ed., Springer, 2017.

[11] L. Zhang and W. Xu, "Encryption Algorithm Security Analysis Using Feature Metrics," Journal of Computer Science and Technology, vol. 32, no. 1, pp. 55–65, 2017.

[12] S. El-Hadedy and A. Hashim, "A Hybrid Cryptography Approach Using ChaCha20 and AES for Enhanced Security," International Journal of Network Security, vol. 18, no. 2, pp. 389–396, 2016.

[13] K. Sayood, Introduction to Data Compression, 5th ed., Morgan Kaufmann, 2016.

[14] C. E. Shannon, "A Mathematical Theory of Communication," Bell System Technical Journal, vol. 27, no. 3, pp. 379–423, 1948.

[15] R. L. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Commun. ACM, vol. 21, no. 2, pp. 120–126, 1978.