International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802007

Deepfake Face Detection Using Machine Learning

Mankanti Hazari Adithya Prasad[1], Gujja Goutham[2], Jangiti Nithish[3], Gurjala Hardhik[4], Mohammad Abdul Rahman[5], Dr. M. Saravanan[6], Dr. Prasad Dharnasi[7]

UG Student, Department of Computer Science and Engineering, Holy Mary Institute of Technology & Science, Telangana, India[1]

UG Student, Department of Computer Science and Engineering, Holy Mary Institute of Technology & Science, Telangana, India[2]

UG Student, Department of Computer Science and Engineering, Holy Mary Institute of Technology & Science, Telangana, India[3]

UG Student, Department of Computer Science and Engineering, Holy Mary Institute of Technology & Science, Telangana, India[4]

UG Student, Department of Computer Science and Engineering, Holy Mary Institute of Technology & Science, Telangana, India[5]

Professor, Holy Mary Institute of Technology & Science, Telangana, India[6]

Professor, Department of Computer Science and Engineering, Holy Mary Institute of Technology & Science, Telangana, India[7]

Publication History: Received: 20.02.2026; Revised: 07.03.2026; Accepted: 10.03. 2026; Published: 15.03.2026.

ABSTRACT: Generative artificial intelligence has evolved rapidly, especially nowadays, allowing to create realistic images with the help of GANS and Diffusion models, which can be used creatively but are a serious threat to forensic integrity and information security, and are often used to commit identity fraud and false information. This research describes full stack web application deepfake face detection that employs a deep learning architecture i.e XceptionNet. However, we have trained two models, XceptionNet and ResNet. But XceptionNet has won as it has an AUC slightly greater than that of ResNet and accuracy. Our project provides full stack web application for users who are non-coders, those who don't know coding, they can also use our platform. Even if they don't know the coding, they can use our platform to detect whether the image is fake or real. The model is trained on data such as a hybrid multi-source dataset strategy using real images from CelebA, Latent Diffusion Models (LFW), and synthetic samples from StyleGAN and stable- diffusion. The experimental evaluation shows that XceptionNet outperforms ResNet152V2 with a testing accuracy of 93%. The system runs on a secure Flask and React.js setup. It uses a RAM-only inference pipeline that protects privacy.

KEYWORDS: Deepfake face Detection, CelebA, StyleGan, Latent Diffusion Models, XceptionNet, ResNet, Area Under the Curve (AUC), Generative Adversarial Networks (GANs), React.js, Flask.

I. INTRODUCTION

In this era of digital, we cannot trust what we see. The AI has developed rapidly more than we can imagine, like it is literally generating the images and videos which are unbelievable, which are more realistic, and is likely impossible for human beings like us to detect that the image is generated by AI and this is just the beginning of AI, we have lot more things to do for every way problem there is at least one solution. Our project proposes a solution to detect whether images are fake or real. In this era, the deepfakes are generated mostly by GANS (Generative Adversarial Network) and Stable Diffusion. GAN mainly follows the 2 parts: Generator and Discriminator, The Generator creates the fake data, whereas Discriminator plays the role of detective, it tries to detect the image and detect whether it's fake or real. They both train and generate the deepfake images. It also uses some techniques like Autoencoders. An autoencoder is a neural network that learns to compress and then reconstruct data. It contains three main parts: Encoder, Bottleneck, and Decoder. For example, there are two people, person A and person B. The encoder learns to compress any face, and there are two decoders: one trained to reconstruct person A’s face, and after training person A’s face, encode it, but decode it using person B’s decoder. The result will be Person B’s face with Person A’s expressions; in simple terms, it is a deepfake image generation process. And there are also many techniques to create deepfake images, such as adding noise to the image.

It is being used as a weapon, as evidenced by the exponential rise in deepfakes used for identity theft. So basically, the AI is now learning at a very fast rate, and so is the technology. To overcome this problem, we proposed our project to classify an image into categories like “fake” or “real”. As these can be used for identity fraud and misbelieved information. Our machine learning model XceptionNet has trained on hybrid datasets, i.e., a good ratio of multiple datasets such as CelebA, StyleGAN, Kaggle 140kFaces Dataset, and LFW Datasets to train the model. As a result, the model has provided very good results as it has categorized the images into real and fake. In this paper, we aim to tackle these challenges by building Full Stack Deep Fake Detection system with the primary objectives as follows:

1.Architectural Comparison: We’ve trained two models and compared side by side XceptionNet and ResNet152V2 for the same dataset. As a result, XceptionNet wins for our project.
2.Generalization: To make sure the system is resistant to various techniques and the models XceptionNet and ResNet are trained on the Hybrid Dataset, such as the combination of CelebA , LFW, and Kaggle140k Faces, which also includes diffusion-generated images, and styleGAN.
3.Secure Deployment: Using a RAM-only processing pipeline, this is done on a full-stack web application built with Flask as the backend and React.js as Frontend for the user-friendly environment. It protects user privacy as it never saves the images uploaded by the user.

II. LITERATURE REVIEW

The developing methods of deepfake detection and the deepfake generation methods of created deepfakes detection created an 'arms race' in the field of deepfake technology detection and creation. To analyze our Full-Stack Deepfake Detection System, we chose eight studies that were most influential in the field. These studies helped us select XceptionNet and ResNet152V2 as the most appropriate architecture(s) to use and justified our choice of building a new custom hybrid dataset (CelebA, StyleGAN, Stable Diffusion) instead of using the old custom datasets.

He et al. [1] presented ResNet, a framework for residual learning, which includes identity (or skip) connections for the successful training of very deep networks and the reduction of optimization degradation, including the vanishing gradient. With these connections, the gradient flows throUGh the skip connections, improving training and feature learning. Relevance: We have used ResNet152V2 as a baseline model to determine if greater architecture depth can assist in the detection of more subtle inconsistencies in facial structures. This model has a large number of parameters, by which results in a longer latency during inference, then we need for an efficiency assessment.

Chollet et al. [2] introduced Xception, a model that substitutes traditional convolutions with depthwise separable convolutions. This method allows for more efficient representation of the spatial and channel-wise correlations. Relevance: we used the Xception model as it is appropriate for identifying fine texture artifacts and color inconsistencies as a result of synthetic image generation, and in our tests, it performed very well and provided the necessary efficiency for real-time processing.

Afchar et al. [3] introduced one of the first deep learning systems for the detection of deepfakes called MesoNet. They focused on mesoscopic features like noise and textures. They showed that manipulating artifacts do not only appear in the high-level semantic structures but also in the intermediate visual features. Relevance: Their research influenced our preprocessing and aUGmentation strategy to enable our models to learn features of artifacts instead of focusing to learn just the face structures.

Yu et al. [4] and Tariq et al. [5] showed that GAN-produced images have subtle artifacts that are generator-specific and are known as artifacts. They are a result of a synthesis and image process. From these findings, it was revealed that detectors that are trained on a single generator do not generalize to other generators. Relevance: We overcome this problem by creating a diverse dataset containing images from various generators, which allows our models to learn generalization for multiple generation methods.

Rössler et al. [6] introduced FaceForensics++, which was the first attempt to create benchmarks to measure the effectiveness of face manipulation detection systems after video sequences are manipulated and compressed. The

FaceForensics++ dataset also helped set the groundwork for how to measure effectiveness for deepfake detection systems. Relevance: learned a lot from the FaceForensics++ dataset, we used other large high-resolution image datasets because the FaceForensics++ dataset is mostly a video dataset.

Li et al. [7] created the Celeb-DF dataset, which helped set benchmarks for evaluating deepfake detection in manipulated videos where the video manipulations were of high quality. Their work showed that many detectors suffer from poor performance at low quality, revealing the low-quality videos and high-quality fake images. Relevance: We aimed to create a collection of real and fake images from many different sources in order to create a dataset that was sufficiently broad for our purposes.

Rombach et al. [8] introduced the first diffusion model-based image synthesis method called Stable Diffusion, which generates images throUGh the iterative process of denoising random noise. As opposed to GAN-based synthesis, diffusion model-based image synthesis creates models with alternative artifacts. Relevance: we use diffusion model-generated images in our dataset to teach our detection system to recognize modern synthesis techniques in addition to GAN-based manipulations.

Verdoliva et al. [9] surveyed media forensics techniques and highlighted limitations of current deepfake detection systems include issues surrounding deployment, privacy, and the overall system design. Relevance: We implement a privacy-preserving detection system that processes images in a way that avoids the use of persistent memory.

III. RESEARCH METHODOLOGY

We used the structured pipeline for frontend and backend integration, and the pipeline is also designed for the model. Such as our model takes input image 299*299 pixels if the input image may contain more or less so the pipelines are designed to convert it into a way where the model accepts and predicts

A. Dataset Collection:

We've collected the dataset manually and created the hybrid dataset the combination of multiple datasets, by balancing real and fake samples. Our total dataset contains approximately 10,000 images, which means our model has trained on 10,000 images, thoUGh it may be fewer. We need the GPUs to train the model, but GPUs are costlier. So, 10,000 images were a good ratio.

- Real Sources: We selected images from CelebA-HQ, Kaggle, and LFW (Labelled Faces in the Wild) datasets for the different qualities of high resolution and lower resolution images
- Fake Sources: We selected images from StyleGAN2 (which creates hyper-realistic images) and StableDiffusion (which has modern fake images)

B. Preprocessing Pipeline:

Raw images cannot be fed directly into a CNN because they vary in size, quality, and format. We introduced a standard preprocessing step using OpenCV to prepare images in a consistent way before passing them to the model.

1.Face Detection: We've utilized Haar Cascade classifiers to locate the face.
2.ROI Cropping: Not all the parts of the image are useful to the model. So, it crops only to the face with 20% more padding for good visualization of the model.
3.Resizing:
-Xception Stream: Resized to 299×299 pixels.
-ResNet Stream: Resized to 224×224 pixels.
4.Normalization: Pixel intensity values were scaled to the range of [0, 1] to speed up convergence.

C. Model Architectures & Training:

We went with a two-phase transfer learning approach for both models:

1. XceptionNet (Primary Model):

Used XceptionNet with pretrained Image weights (Transfer Learning)
Phase 1- Kept XceptionNet fully frozen, only trained the custom dense layers + attention mechanism
Phase 2- Unfreeze the top layers (kept first 100 layers frozen), fine-tuning with 10x smaller learning rate (0.00001) for 40 epochs

2. **ResNet152V2:**
Backbone: ResNet152v2 is the 152-layered backbone.
-It can learn complex features with the aid of residual skip connections.
Protocol for Training:
We train the classification head for 15 epochs while freezing the base layers, and in the next phase, we use the AdamW optimizer (learning rate: 1e-5) [10]. We then unfreeze top 30% of layers, and then we adjust it to focal loss for the new challenging cases.

D. Full Stack Architecture:

This system is a fast, secure, and user-friendly web-based platform.
ML Model:

The model analyzes facial images and predicts the probability of an image being real or fake. (score 0-1)
Two-phase training: In the first phase, train the classification head while keeping the Xception base frozen, and then unfreeze later layers and fine-tune — this yield stable, and accurate results. Architecture highlights: Xception base (pretrained), attention map to focus on eyes/mouth/boundaries, then dense and dropout layers to decide whether real or fake. Loss and optimizer: Focal Loss is used for class imbalance, and AdamW (with weight decay) is used for stability and efficiency.
L2 regularization, and callbacks (Checkpoint, ReduceLR, EarlyStopping, TensorBoard, etc.) to avoid overfitting and for the best model.

Backend (inference & API):

Minimal and Quick: Flask service reduces extra work for every request by running the .h5 model only once during startup.
Simple architecture: POST → OpenCV face detection and crop → resize (PIL), fix for 299x299 → normalize → model.predict → return JSON (label + confidence).
Frontend communication: Axios sends images as Form data, and the backend executes browser API calls via CORS.
-It gives the response in the JSON format, for example:

```
{
 "prediction": "FAKE",
 "confidence": 92.53,
 "raw_score": 0.0242,
 "face_detected": true
}
```

Frontend (user flow & UX):

In Our Project, we used React.js as the frontend in our full-stack web application. React.js is a library for single-page web application. Axios is the bridge for the frontend to connect to the frontend and backend it takes the uploaded image from the user and sends it to Flask, and gets a JSON response from it.

```
const res = await axios.post('http://localhost:5000/predict', formData);
setResult(res.data); // receives the JSON
```

Code which connects Frontend and Backend
We've used modular components (Navbar, Scanner, and Result) for reusability, and react-router for pages (Home, About, and Contact). UseState stores file loading and user information so that the user interface feels responsive.

Computer vision (data processing and face detection):

Finding a face: Using OpenCV, we begin with a Haar cascade to locate the largest face in the image. Then we crop to the face with a 20 percent padding to ensure we do have edge artifacts missed.
Detection grayscale: Subsequently, using grayscale detection improves speed and efficiency to tackle variations in lighting.
Detect Multi Scale parameters: We tune the scaleFactor and minNeighbors to find a balance between scale invariance and false positives.
Importance of padding: Many deepfake art faces have padding issues, so it catches the issues with deepfake artifacts that border the padding that models typically miss.
-Overall, using CV models helps the system in solving key pixel-only issues in a specific area, and this helps to greatly improve the performance and consistency of the system.

IV. RESULTS AND DISCUSSION

In-depth data was obtained by comparing the two architectures.

A. Comparative Performance Analysis:

TABLE I: COMPARATIVE ANALYSIS OF DEEP LEARNING ARCHITECTURES

Evaluation Metric	**XceptionNet (Proposed)**	**ResNet152V2 (Baseline)**
Test Accuracy	93.20%	91.50%
AUC Score	0.9848	0.9841
Precision	91.54%	89.69%
Recall (Sensitivity)	95.20%	93.80%
F1-Score	93.33%	91.70%
Total Parameters	22.9 million	59.5 million
Model Weight Size	88 MB	235 MB
Inference Time (CPU)	~0.18 sec	~0.45 sec

TABLE II: DETAILED PERFORMANCE METRICS OF XCEPTIONNET

Performance Metric	**Value**	**Detection Significance**
Overall Test Accuracy	93.20%	High reliability for general usage.

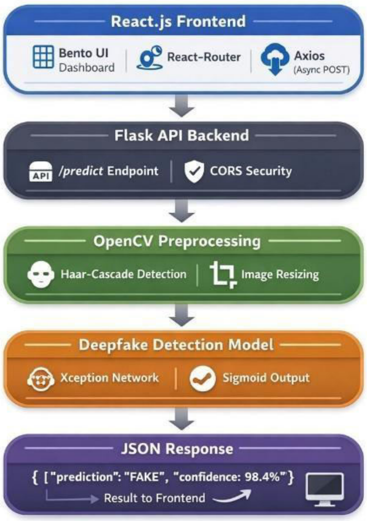
True Positive Rate (Recall - FAKE)	95.20%	Critical: Captures 95% of all deepfakes; very low miss rate.
True Negative Rate (Recall - REAL)	91.20%	Effectively identifies authentic images, reducing false alarms.
False Negative Count	24 / 500	Only 24 fake images out of 500 managed to fool the system.
Training Accuracy (Phase 2)	93.20%	Performance after fine-tuning specifically for GAN artifacts.

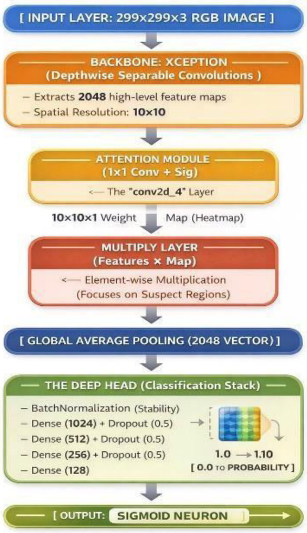
B. System Capabilities & Full Stack Advantages

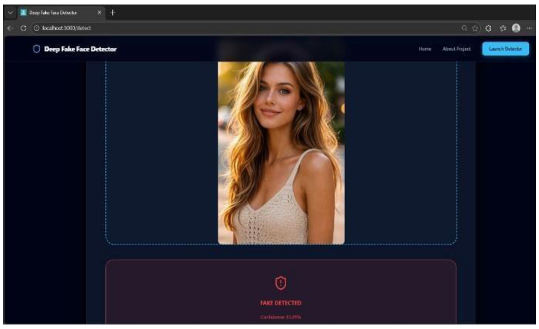
1.Reduced Latency: System performance was directly impacted by XceptionNet's parameter efficiency (22.9M vs. 59.5M). The application's fast feedback made it possible as its inference time is approximately 0.18 seconds, which is a crucial aspect of user experience.

2.Improved Security and Privacy: A significant ethical issue is solved by the "ZeroRetention" privacy policy built into the Flask backend. The system reduces the attack surface and guarantees adherence to data protection standards by processing data in volatile memory (RAM) and avoiding disk writes.

3.Scalability: Because the model is small (88 MB), it can be used with inexpensive cloud instances without the need for pricey GPU inference clusters, which lowers operating costs by about 40%.

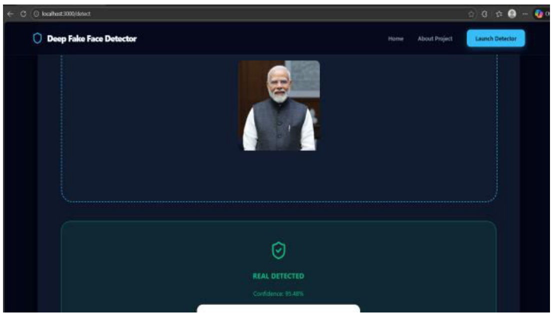



[INPUT LAYER: 299x299x3 RGB IMAGE]
BACKBONE: XCEPTION
(Depthwise Separable Convolutions)
– Extracts 2048 high-level feature maps
– Spatial Resolution: 10x10
ATTENTION MODULE
(1x1 Conv + Sig)
<— The "conv2d_4" Layer
10×10×1 Weight
Map (Heatmap)
MULTIPLY LAYER
(Features × Map)
<— Element-wise Multiplication
(Focuses on Suspect Regions)
[GLOBAL AVERAGE POOLING (2048 VECTOR)]
THE DEEP HEAD (Classification Stack)
– BatchNormalization (Stability)
– Dense (1024) + Dropout (0.5)
– Dense (512) + Dropout (0.5)
– Dense (256) + Dropout (0.5)
– Dense (128)
1.0 → 1.10
[0.0 to PROBABILITY]
[OUTPUT: SIGMOID NEURON



FAKE DETECTED

Confidence: 93.85%



REAL DETECTED

Confidence: 95.48%

V. CONCLUSION

We cannot easily trust the things we see on digital, and the risk of trusting the digital would be high. Our project has developed a solution to detect whether images are fake or real. And our project has developed a good full-stack web application that is quick, easy, secure, and user-friendly. We concluded that the XceptionNet is the best architecture for the problem statement or the task, outperforming the ResNet152V2 model with 93.2% accuracy and 98.4% AUC. ThroUGh the use of an Xception backbone, spatial attention mechanisms, and multi-source training.

FUTURE WORK

While the current system produces excellent results for image-based deepfakes, there are multiple areas that can be improved:

1.Video Sequence Analysis: The pipeline can be extended to accept video uploads, and temporal problems such as flickering faces can be addressed by adding Recurrent Neural Networks (LSTM) or frame-averaging logic.
2.Ensemble Learning: Combining Xception and ResNet152V2 architectures to exploit the feature extraction capabilities of both may improve accuracy above 95%, althoUGh Xception performed better on its own.
3.Explainable AI (XAI): Adding Grad-CAM (Gradient-weighted Class Activation Mapping) to the frontend, it can reveal and highlight the areas of why the model labeled an image as fake or real. which helps to build user confidence.
4.Mobile Application: For online or offline detection on Android and iOS devices, we can use TensorFlow Lite (TFLite) to deploy the optimized Xception model.

REFERENCES

1. Vani, S., Malathi, P., Ramya, V. J., Sriman, B., Saravanan, M., & Srivel, R. (2024). An efficient black widow optimization-based faster R-CNN for classification of COVID-19 from CT images. Multimedia Systems, 30(2), 108.
2. Kumar, A. S., Saravanan, M., Joshna, N., & Seshadri, G. (2019). Contingency analysis of fault and minimization of power system outage using fuzzy controller. International Journal of Innovative Technology and Exploring Engineering, 9(1), 4111-4115.
3. David, A. (2020). Air pollution control monitoring & delivery rate escalated by efficient use of markov process in manet networks: to measure quality of service parameters. Test Engineering & Management, The Mattingley Publishing Co., Inc. ISSN, 0193-4120.
4. Saravanan, M., Kumar, A. S., Devasaran, R., Seshadri, G., & Sivaganesan, S. (2019). Performance analysis of very sparse matrix converter using indirect space vector modulation. Intern. Jou. of Inn. Techn. and Expl. Eng, 9(1), 4756-4762.
5. Saravanan, M., & Sivakumaran, T. S. (2016). Three phase dual input direct matrix converter for integration of two AC sources from wind turbines. Circuits Syst., 7, 3807-3817.
6. Dharnasi, P. (2025). A Multi-Domain AI Framework for Enterprise Agility Integrating Retail Analytics with SAP Modernization and Secure Financial Intelligence. International Journal of Humanities and Information Technology, 7(4), 61-66.
7. Amitha, K., Ram Manohar Reddy, M., Yashwanth, K., Shylaja, K., Rahul Reddy, M., Srinu, B., & Dharnasi, P. (2026). AI empowered security monitoring system with the help of deployed ML models. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(1), 69–73.

8. Gogada, S., Gopichand, K., Reddy, K. C., Keerthana, G., Nithish Kumar, M., Shivalingam, N., & Dharnasi, P. (2026). Cloud computing/deep learning customer churn prediction for SaaS platforms. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(1), 74–78.
9. Akula, A., Budha, G., Bingi, G., Chanda, U., Borra, A. R., Yadav, D. B., & Saravanan, M. (2026). Emotion recognition from facial expressions using CNNs. International Journal of Engineering & Extended Technologies Research (IJEETR), 8(1), 120–125.
10. Varshini, M., Chandrapathi, M., Manirekha, G., Balaraju, M., Afraz, M., Sarvanan, M., & Dharnasi, P. (2026). ATM access using card scanner and face recognition with AIML. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(1), 113–118.
11. Feroz, A., Pranay, D., Srikar Sai Raj, B., Harsha Vardhan, C., Rohith Raja, B., Nirmala, B., & Dharnasi, P. (2026). Blockchain and machine learning combined secured voting system. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(1), 119–124.
12. Tirupalli, S. R., Munduri, S. K., Sangaraju, V., Yeruva, S. D., Saravanan, M., & Dharnasi, P. (2026). Blockchain integration with cloud storage for secure and transparent file management. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(1), 79–86.
13. Chandu, S., Goutham, T., Badrinath, P., Prashanth Reddy, V., Yadav, D. B., & Dharnas, P. (2026). Biometric authentication using IoT devices powered by deep learning and encrypted verification. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(1), 87–92.
14. Singh, K., Amrutha Varshini, G., Karthikeya, M., Manideep, G., Sarvanan, M., & Dharnasi, P. (2026). Automatic brand logo detection using deep learning. International Journal of Engineering & Extended Technologies Research (IJEETR), 8(1), 126–130.
15. Keerthana, L. M., Mounika, G., Abhinaya, K., Zakeer, M., Chowdary, K. M., Bhagyaraj, K., & Prasad, D. (2026). Floods and landslide prediction using machine learning. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(1), 125–129.
16. Dadigari, M., Appikatla, S., Gandhala, Y., Bollu, S., Macha, K., & Saravanan, M. (2026). Bitcoin price prediction with ML through blockchain technology. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(1), 130–136.
17. Chinthala, S., Erla, P. K., Dongari, A., Bantu, A., Chityala, S. G., & Saravanan, M. S. (2026). Food recognition and calorie estimation using machine learning. International Journal of Engineering & Extended Technologies Research (IJEETR), 8(2), 480–488.
18. Chinthamalla, N., Anumula, G., Banja, N., Chelluboina, L., Dangeti, S., Jitendra, A., & Saravanan, M. (2026). IoT-based vehicle tracking with accident alert system. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(2), 486–494.
19. Nagamani, K., Laxmikala, K., Sreeram, K., Eshwar, K., Jitendra, A., & Dharnasi, P. (2026). Disaster management and earthquake prediction system using machine learning. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(2), 495–499.
20. Prasad, E. D., Sahithi, B., Jyoshnavi, C., Swathi, D., Arun Kumar, T., Dharnasi, P., & Saravanan, M. (2026). A technology driven – solution for food and hunger management. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(2), 440–448.
21. Rakesh, V., Vinay Kumar, M., Bharath Patel, P., Varun Raj, B., Saravanan, M., & Dharnasi, P. (2026). IoT-based gas leakage detector with SMS alert. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(2), 449–456.
22. Chanamalla, B., Murali, V. N., Suresh, B., Deepak, M. S., Zakriya, M., Yadav, D. B., & Saravanan, M. (2026). AI-driven multi-agent shopping system through e-commerce system. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(2), 463–470.
23. Bhagyasri, Y., Bhargavi, P., Akshaya, T., Pavansai, S., Dharnasi, P., & Jitendra, A. (2026). IoT based security & smart home intrusion prevention system. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(2), 457–462.
24. Thotla, S. B., Vyshnavi, S., Anusha, P., Vinisha, R., Mahesh, S., Yadav, D. B., & Dharnasi, P. (2026). Traffic congestion prediction using real time data by using deep learning techniques. , 8(2), 489–494.
25. Rupika, M., Nandini, G., Mythri, M., Vasu, K., Abhiram, M., Shivalingam, N., & Dharnasi, P. (2026). Electronic gadget addiction prediction using machine learning. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(2), 500–505.
26. Akshaya, N., Balaji, Y., Chennarao, J., Sathwik, P., & Dharnasi, P. (2026). Diabetic retinopathy diagnosis with deep learning. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(2), 506–512.

27. Pavan Kumar, T., Abhishek Goud, T., Yogesh, S., Manikanta, V., Dinesh, P., Srinu, B., & Dharnasi, P. (2026). Smart attendance system using facial recognition for staff using AI/ML. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(2), 513–519. https://doi.org/10.15662/IJRPETM.2026.0902005
28. Reddy, V. N., Rao, P. H. S., Singh, N. S., Kumar, V. S. S., Reddy, Y. B., & Dharnasi, P. (2026). Face recognition using criminal identification system. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(2), 520–527.
29. Rachana, P., Kalyan, P. P., Kumar, T. S., Reddy, P. M., Rohan, P., Saravanan, M., & Dharnasi, P. (2026). Secure chat application with end-to-end encryption using deep learning. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(2), 472–478.
30. Krishna, G., Rajesh, B., Dinesh, B., Sravani, B., Rajesh, G., Dharnasi, P., & Sarvanan, M. (2026). Smart agriculture system using IoT with help of AI-techniques. International Journal of Computer Technology and Electronics Communication, 9(2), 479–487.
31. Reddy, N. H. V., Reddy, N. T., Bharath, M., Hemanth, N., Dharnasi, D. P., Nirmala, B., & Jitendra, A. (2026). AI based learning assistant using machine learning. International Journal of Engineering & Extended Technologies Research, 8(2), 495–504.
32. Vangara, N., Bhargavi, P., Chandu, R., Bhavani, V., Yadav, D. B., & Dharnasi, P. (2026). Machine learning based intrusion detection system using supervised and unsupervised learning. International Journal of Engineering & Extended Technologies Research (IJEETR), 8(2), 505–511.
33. Yadamakanti, S., Mahesh, Y., Rathnam, S. A., Praveen, V., Jitendra, A., & Dharnasi, P. (2026). Unified Payments Interface fraud detection using machine learning. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(2), 488–497.
34. Basha, S. A., Krishna, V. S. B., Shanker, S. S., Sravya, R., Shivalingam, N., & Dharnasi, P. (2026). AI-powered price prediction for agriculture markets. International Journal of Engineering & Extended Technologies Research (IJEETR), 8(2), 512–515.
35. Sanjay, P., Vardhan, Y. H., Raja, S. Y., Krishna, V. M., Nirmala, B., & Dharnasi, P. (2026). Disaster management and earthquake tsunami prediction system using machine learning and deep learning. International Journal of Engineering & Extended Technologies Research (IJEETR), 8(2), 516–522.
36. Varsha, P., Chary, P. K., Sathvik, P., Varma, N. V., Rahul, S., Saravanam, M., & Dharnasi, P. (2026). IoT-based fire alarm and location tracking system. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 9(2), 528–532.
37. Priya, B. A., Gayathri, D., Maheshwari, B., Nikhitha, C., Sravanam, D., Yadav, D. B., & Saravanan, M. (2026). Fake news detection using natural language processing. International Journal of Computer Technology and Electronics Communication (IJCTEC), 9(2), 498–505.
38. C.Nagarajan and M.Madheswaran - ‘Performance Analysis of LCL-T Resonant Converter with Fuzzy/PID Using State Space Analysis’- Springer, Electrical Engineering, Vol.93 (3), pp.167-178, September 2011.
39. Gowtham, M. S., Ramkumar, M., Jamaesha, S. S., & Vigenesh, M. (2024). Artificial self-attention rabbits battle royale multiscale network based robust and secure data transmission in mobile Ad Hoc networks. Computers & Security, 142, 103889.
40. Sivanantham, E., Vijayakumar, R., Veda, P., Nithya, A., Vinayagam, P. V., & Renukadevi, S. (2024, April). Optimizing Smart Methane Farms: Intelligent Waste Sorting for Maximum Biogas Yield through Naive Bayes and IoT Integration. In 2024 10th International Conference on Communication and Signal Processing (ICCSP) (pp. 1205-1210). IEEE.