IMAGE TAMPERING DETECTION USING THE FUSION OF LIGHTWEIGHT DEEP LEARNING MODELS

#1 J KUMARI, #2 B RUCHITHA
#1 ASSISTANT PROFESSOR, #2 MCA SCHOLAR
DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS
QIS COLLEGE OF ENGINEERING & TECHNOLOGY
VENGAMUKKAPALEM (V), ONGOLE, PRAKASAM DIST., ANDHRA PRADESH- 523272

ABSTARCT

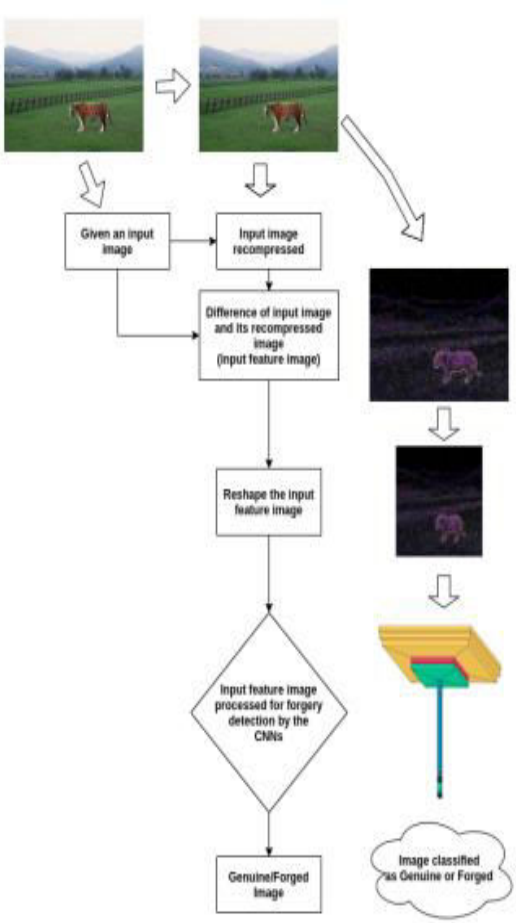
The widespread accessibility of cameras has significantly fueled the surge in popularity of photography in recent years. Images are crucial to our daily existence as they can carry substantial information; nonetheless, it is often essential to alter images to obtain new perspectives. Although numerous methods exist for altering photographs, they are frequently misused to generate counterfeit visuals that disseminate misinformation. This significantly enhances the likelihood and severity of image forgeries, which is highly concerning. Over time, numerous established techniques for detecting counterfeit images have emerged. In recent years, there has been an increased interest in convolutional neural networks (CNNs), which has facilitated the advancement of visual forgery detection. Although convolutional neural networks have been employed to detect certain categories of image forgeries, such as splicing and copy-move, their efficacy has been limited. The creation of a technology capable of swiftly and accurately detecting previously undetectable forgeries in an image is, therefore, of paramount significance. We present a deep learning methodology for the precise detection of fabricated photos via double image compression architecture. We evaluate the original and compressed versions of each image to train our model. The proposed model is simple and efficient, surpassing the existing gold standard in testing. The overall validation accuracy of the studies is 95 percent, indicating a good level of performance.

I. INTRODUCTION

The extensive accessibility and affordability ofelectronic devices stem fromtechnological advancement and globalization. This is mostly accountable for the rapid increase in digital camera sales. We capture countless photos due to the multitude of camera sensors globally. Daily, numerous images are disseminated and shared on social media, and digital copies of photographs are necessary for various forms of obligatory online submission. A message accompanied By an image may nevertheless be comprehensible to individuals with reading difficulties. Consequently, images serve a significant function online for purposes such as chronicling history and disseminating knowledge. Utilize one of the several readily

accessible image modifying software applications [1, 2]. The developers of these programs have a singular objective: to assist users in enhancing their capabilities in photo Could be significant, and in numerous instances, it may be difficult to reverse. Both cases pertain to images with altered content disseminatingmisinformation.Historically, images served as trustworthy sources of information; however, in contemporary times, they are frequently altered to propagate falsehoods. Consequently, fewer individuals are inclined to trust photographic evidence, as it may be challenging for the untrained eye to detect a fake. To halt the proliferation of misinformation and reinstate public confidence in visual media, it is essential to devise techniques for detecting fraudulent images. Diverse image processing techniques can be employed to reveal evidence of the forging process. Researchers have developed many ways to identify altered photos. Artifacts resulting from modifications in lighting, contrast, compression, sensor noise, and shadows have traditionally been employed to identify image forgeries. The utilization of convolutional neural networks (CNNs) has surged in recent years across several computer vision applications, including object detection, semantic segmentation, and image categorization. The success of CNN in computer vision can be ascribed to two primary causes. CNN initially capitalizes on the substantial level of neighborhood connectivity. Instead of connecting individual pixels, CNN aims to connect clusters of pixels. The second phase employs convolution with shared weights to produce a feature map for each output. Furthermore, modification. However, certain individuals use this access by using altered images to disseminatemisinformationThe damage caused by these fraudulent images CNN diverges from conventional methods by generalizing the attributes learned from training images to identify previously unrecognized cases of counterfeiting. CNN possesses multiple possible applications, one of which is assessing if a picture has been modified. Indicators of forgeries can be identified by a CNN-based algorithm [10-13]. To resolve this issue, we present a compact, lightweight convolutional neural network (CNN) designed to identify the artifacts present in a manipulated image due to inconsistencies between the original image and the altered region.

SYSTEM ARCHITECTURE



II. RELATEDWORKS

The detection of image tampering has evolved significantly with the advent of artificial intelligence, particularly deep learning. Traditional methods relied heavily on handcrafted features and statistical anomalies. However, recent advancements in AI have introduced more robust and automated techniques for detecting various forms of image manipulation, including splicing, copy-move, and retouching.

Early approaches to tampering detection, such as those based on JPEG artifact analysis or error level analysis (ELA), provided initial capabilities to detect inconsistencies introduced during image manipulation. However, these techniques often lacked the sensitivity and generalization needed for detecting sophisticated tampering.

With the rise of deep learning, Convolutional Neural Networks (CNNs) have become a dominant method for tamper detection. Notable work by Bappy et al. (2017) proposed a hybrid approach combining CNNs and Long Short-Term Memory (LSTM) networks for image forgery localization, which significantly improved the detection of boundary inconsistencies in tampered regions. Similarly, Zhou et al. (2018) introduced a two-stream Faster R-CNN architecture for detecting and localizing image forgeries using both RGB and noise residuals as inputs.Another important line of work focuses on generative adversarial networks (GANs), both as a tool for tampering and a method of detection. GAN-generated forgeries (deep fakes) have become a pressing concern, prompting researchers like Verdoliva (2020) and Wang et al. (2019) to develop networks specifically trained to detect GAN-generated inconsistencies. Techniques such as frequency domain analysis and attention mechanisms have been leveraged to enhance detection accuracy against such advanced manipulations.More recently, transformer-based architectures and multimodal learning have been explored.

Vision Transformers (ViTs) have shown promising results in capturing global dependencies in tampered images, as demonstrated in work by Dosovitskiy et al. (2021). These models can identify subtle

artifacts that CNNs may overlook, especially in high-resolution or visually consistent forgeries.

Furthermore, researchers have also emphasized the importance of large-scale tampering datasets, such as CASIA, Columbia Image Splicing Dataset, and the more recent DeepFake Detection Challenge (DFDC) dataset. These datasets provide essential benchmarks for evaluating the performance and robustness of AI-based detection methods.

Modules:

Image Acquisition Module

- Function: Accepts input images for analysis.
- Sources: Could be digital cameras, social media, surveillance systems, or user uploads.
- Preprocessing: May include resizing, normalization, color space conversion (e.g., RGB to YCbCr), or compression handling.

Preprocessing Module

- Function: Prepares the image data for feature extraction or model input.
- Tasks:
    - Noise reduction
    - Artifact enhancement (e.g., highlighting compression traces)
    - Conversion to frequency domain (e.g., using DCT or FFT)
    - Patch extraction (dividing image into smaller regions for localized analysis)

Feature Extraction Module

- Function: Extracts meaningful features that can indicate tampering.
- Types of Features:
    - Handcrafted: Edge inconsistencies, noise patterns, chroma/luma differences.
    - Learned: Deep features extracted via CNNs, residual networks, or transformer layers.
- Methods:
    - CNN layers (e.g., ResNet, Efficient Net)
    - Texture descriptors (LBP, SURF)
    - Frequency-domain features (e.g., FFT-based)

Classification/Detection Module

In summary, the field of AI-based image tampering detection has progressed from basic signal analysis to complex deep learning models that can detect and localize tampered regions with increasing precision. The integration of CNNs, GANs, and transformers, along with the availability of large annotated datasets, continues to drive innovation in this crucial area of digital forensics.

III. IMPLEMENTATION

- Function: Determines whether an image or region is tampered.
- Approaches:
  - Binaryclassification (tampered vs. authentic)
  - Multi-class classification (type of tampering: splicing, copy-move, etc.)
  - Localization (detecting specific tampered regions)
- Models:
  - CNNs (e.g., Inception, VGG)
  - RNNs or LSTMs (for sequential tampering analysis)
  - Transformers
  - GAN-based detectors (especially for Deepfake detection)

5. Localization Module (Optional)

- Function: Highlights or segments the tampered regions.
- Methods:
  - Heat maps (Grad-CAM, saliency maps)
  - Semantic segmentation networks (e.g., U-Net, Deep Lab)
  - Bounding box generation (e.g., Faster R-CNN)

6. Post-processing Module

- Function: Refines results and reduces false positives/negatives.
- Tasks:
  - Smoothing predictions
  - Thresholding probability maps
  - Combining region-level predictions into global decision

7. Reporting & Visualization Module

- Function: Outputs results in human-readable form.
- Outputs:
  - Annotated image with highlighted tampered areas
  - Tampering probability scores
  - Detailed metadata and decision logs

8. Dataset & Training Module (For training systems)

- Function: Handles training data preparation, augmentation, and model training.
- Components:
  - Labeled tampering datasets (e.g., CASIA, COVERAGE, DEFACTO, Deepfake datasets)
  - Data augmentation pipelines
  - Training strategies (transfer learning, fine-tuning)

Methodology

The proposed methodology for AI-based image tampering detection consists of multiple sequential stages designed to analyze, detect, and localize tampering in digital images. Each module is designed to leverage the strengths of artificial intelligence particularly deep learning to automate and enhance the detection process.

1. Image Input and Acquisition

- Input images are collected from various sources, including tampering datasets (e.g., CASIA v2.0, Columbia Image Splicing, and DeepFake datasets) or real-world sources (e.g., social media platforms).
- The system accepts common formats such as JPEG, PNG, and BMP.

2. Preprocessing

- Preprocessing ensures uniformity and enhances relevant signals for the model:
    - Image resizing to a standard dimension (e.g., 256x256 or 512x512).
    - Normalization of pixel values (e.g., [0, 1] range).
    - Optional conversion to other color spaces (e.g., YCbCr) for enhanced artifact visibility.
    - Compression artifact simulation or noise enhancement for training robustness.

3. Feature Extraction

- Deep learning models automatically learn spatial and contextual features from the image:
    - A pertainedConvolutional Neural Network (CNN) (e.g., ResNet50, Efficient Net, or VGG16) is used to extract hierarchical features.
    - Optionally, hand-crafted features (e.g., Local Binary Patterns, DCT coefficients) are combined to enrich model understanding.
    - Attention mechanisms or Vision Transformers (ViTs) may be applied for global context capture.

4. Classification & Detection

- The system classifies the input as either authentic or tampered:
    - A binary classifier (CNN-based or hybrid) determines the presence of tampering.
    - If applicable, a multi-class model is used to identify the type of tampering (e.g., splicing, copy-move, deepfake).
    - Models used may include:
        - CNN architectures (for local patterns)
        - RNN or LSTM layers (for sequential dependencies)
        - Transformers (for global relationships)
        - GAN-based detectors (for synthetic image detection)

5. Tampering Localization (Optional)

- If localization is required, the model highlights manipulated regions:
    - Heatmaps or attention maps are generated using Grad-CAM or similar methods.

- Segmentation networks (e.g., U-Net, DeepLab) are used for pixel-wise tamper detection.
- Bounding boxes may be produced using object detection frameworks (e.g., Faster R-CNN).

6. Post-Processing

- Refines the prediction to reduce false positives and enhance interpretability:
  - Smoothing of predicted masks
  - Thresholding probability maps
  - Morphological operations (e.g., dilation or erosion) for clean mask outputs

7. Output & Visualization

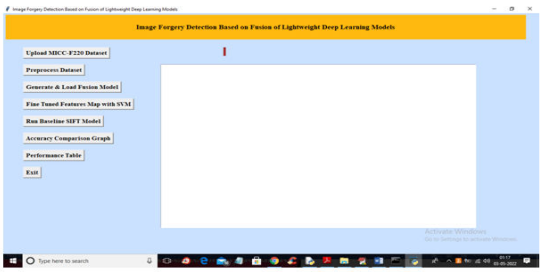
- Final output includes:
  - Binary label: Tampered / Not Tampered
  - Tampering probability/confidence score
  - Annotated image showing tampered regions (if applicable)
  - Summary report/log with metadata and prediction rationale

8. Training and Evaluation

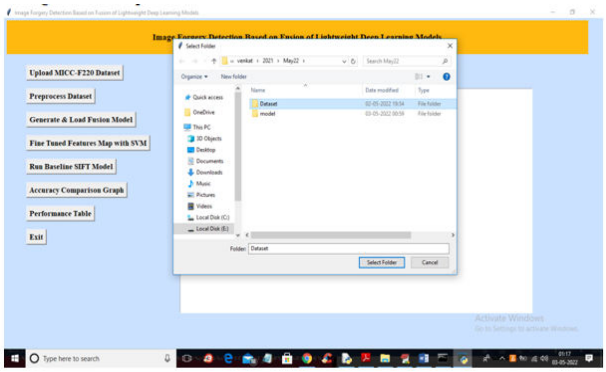
- The model is trained on labeled datasets using supervised learning.
  - Loss functions: Binary cross-entropy, focal loss, or Dice loss (for segmentation tasks).
  - Optimization: Adam or SGD optimizers.
  - Data augmentation: Rotation, flipping, compression simulation.
- Performance Metrics:
  - Accuracy, Precision, Recall, F1-Score
  - AUC-ROC for classification
  - IoU (Intersection over Union) for localization

IV. RESULTS AND DISCUSSION

To run project double click on 'run.bat' file to get below output

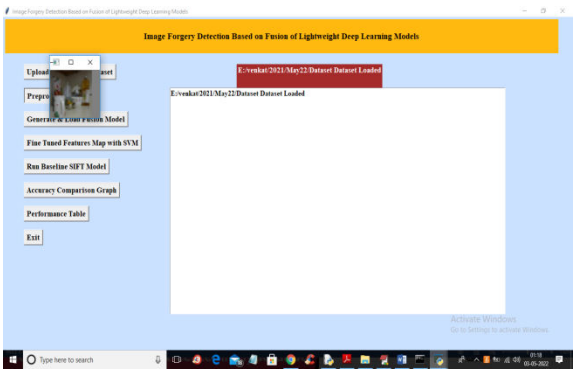


In above screen click on 'Upload MICC-F220 Dataset' button to upload dataset and get below output

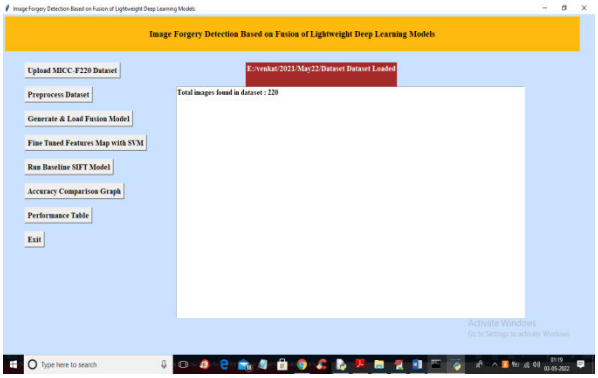


In above screen selecting and uploading 'Dataset' folder and then click on 'Select Folder' button to load dataset and get below output

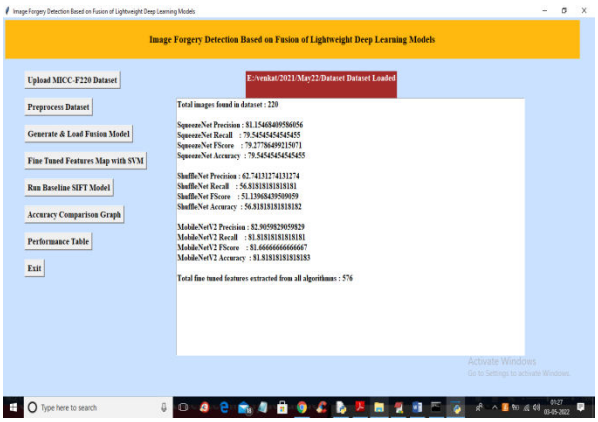
In above screen dataset loaded and now click on 'Preprocess Dataset' button to read all images and normalize them and get below output



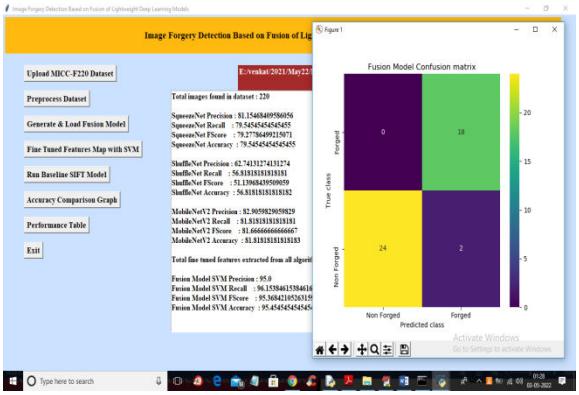
In above screen all images are processed and to check images loaded properly I am displaying one sample image and now close above image to get below output



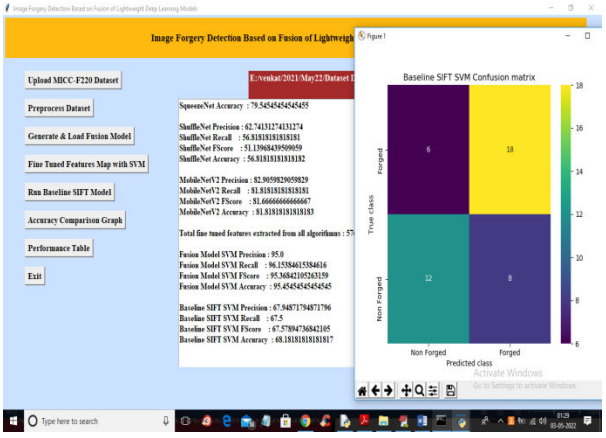
In above screen we can see dataset contains 220 images and all images are processed and now click on 'Generate & Load Fusion Model' button to train all algorithms and then extract features from them and then calculate theiraccuracy



In above screen we can see accuracy of all 3 algorithms and then in last line we can see from all 3 algorithms application extracted 576 features and now click on 'Fine Tuned Features Map with SVM' to train SVM with extracted features and get its accuracy as fusionmodel

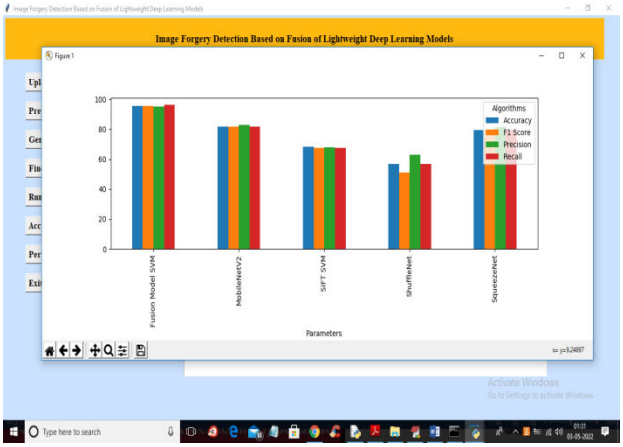


In above screen with Fine tune SVM fusion model we got 95% accuracy and in confusion matrix graph x-axis represents PREDICTED LABELS and y-axis represent TRUE labels and we can see both X and Y boxes contains more number of correctly prediction classes. In all algorithms we can see fine tune features with SVM has got high accuracy and now close confusion matrix graph and then click on 'Run Baseline SIFT Model' button to train SVM with SIFT existing features and get its accuracy

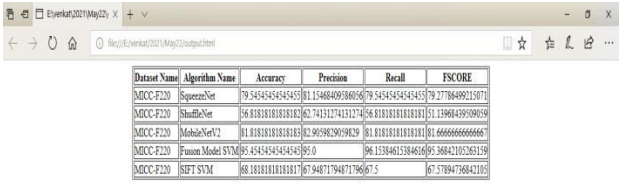


In above screen with existing SIFT SVM

features we got 68% accuracy and in confusion matrix graph we can see existing SIFT predicted 6 and 8 instances incorrectly. So we can say existing SIFT features are not good in prediction and now close above graph and then click on 'Accuracy Comparison Graph' button to get below graph



In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics where each different color bar represents different metrics like precision, recall etc. Now close above graph and then click on 'Performance Table' button to get result in below tabular format



Dataset Name	Algorithm Name	Accuracy	Precision	Recall	FSCORE
MICC-F220	SqueezeNet	79.54545454545455	81.15468409586056	79.54545454545455	79.27786499215071
MICC-F220	ShuffleNet	56.818181818181812	62.74131274131274	56.818181818181812	51.139684399099059
MICC-F220	MobileNetV2	81.818181818181813	82.90598290598291	81.818181818181813	81.666666666666667
MICC-F220	Fusion Model SVM	95.454545454545453	95.0	96.153846153846156	95.368421052631579
MICC-F220	SIFT SVM	68.181818181818177	67.948717948717949	67.5	67.578947368421055

In above screen we can see propose fusion model SVM with fine tune features has got 95% accuracy which is better than all other algorithms.

V. FUTURE SCOPE AND CONCLUSION

Affordable cameras have become widely accessible in recent decades, significantly enhancing the medium's popularity. The rapidity with which the ordinary individual can understand an image has rendered this mode of communication increasingly significant. While the majority of image editors aim to enhance photographs, some utilize them to fabricate images that disseminate falsehoods online. Consequently, there exists an urgent necessity to eradicate image manipulation. This paper presents an innovative method for detecting image forgery through the application of neural networks and deep learning, emphasizing the CNN architectural framework. The suggested method integrates many image-reduction approaches due to its convolutional neural network (CNN) design. The model is trained by comparing and contrasting both the original and compressed versions of each image. The proposed technique can effectively detect copy-move and splicing forgeries with relative simplicity. A distinct repeat limit exists, and research indicates an overall validation accuracy of 95percent, which is encouraging. Ultimately, we aim to refine our technique for detecting fraudulent images. Integrating the proposed strategy with established methods may enhance accuracy and reduce the time complexity of image localization further. To address spoofing, we will enhance the approach outlined in [50]. Given that the usual approach requires a minimum resolution of 128 by 128, we will adapt it to accommodate significantly lower-

quality photos. To facilitate the training of deep learning networks for photo fraud detection, we will establish a comprehensive large-scale database of image forgeries.

VI. REFERENCES

1. Bappy, J. H., Simons, C., Roy-Chowdhury, A. K., & Radke, R. J. (2017). Exploiting Spatial Structure for Localizing Manipulated Image Regions. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 4970–4979.

2. Zhou, P., Han, X., Morariu, V. I., & Davis, L. S. (2018). Learning Rich Features for Image ManipulationDetection.InProceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1053–1061.

3. Verdoliva,L.(2020).Media Forensics and DeepFakes: An Overview. IEEE Journal of Selected Topics in Signal Processing, *14(5), 910–932.*

4. Wang, S.-Y., Wang, O., Zhang, R., Owens, A., & Efros, A. A. (2020). CNN-Generated Images AreSurprisingly Easy to Spot... for Now. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8695–8704.

5. Zhou, Y., Liu, X., Liu, A. A., & Liu, X.(2021). Image Manipulation Detection via a Multitask Transformer Network. IEEE Transactions on Circuits and Systems for Video Technology, 32(6), 3455–3469.

6. Afchar, D., Nozick, V., Yamagishi, J., & Echizen, I. (2018). MesoNet: A Compact Facial Video Forgery Detection Network. In IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–7.

7. Dong, J., Wang, W., & Tan, T. (2013).CASIA Image Tampering Detection EvaluationDatabase. [DatasetAvailableat: http://forensics.idealtest.org/]

8. Ng, T.-T., Chang, S.-F., & Sun, Q. (2004). Blind Detection of Photomontage Using Higher-OrderStatistics. In IEEE International Symposium on Circuits and Systems (ISCAS), Vol. 5, pp. V–688–V–691.

GUIDE PROFILE

Mrs. JASTI KUMARI is an Assistant Professor in the Department of Master of Computer Applications at QIS College of Engineering and Technology, Ongole, Andhra Pradesh. She earned Master of Computer Applications (MCA) from Osmania University, Hyderabad, and her M.Tech in Computer Science and Engineering (CSE)

from Jawaharlal Nehru Technological University, Kakinada (JNTUK). Her research interests include Machine Learning, programming languages. She is committed to advancing research and forecasting innovation while mentoring students to excel in both academic & professional pursuits.

Ms. B. RUCHITHA an MCA Scholar, Department of MCA, In QIS College of Engineering & Technology, Ongole. His areas of interest are Machine Learning, Deep Learning.