International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Image Forgery Detection Using Attention-Aware Hierarchical-Feature Fusion Module Approach

B. Pravallika[1*], Sasikala Jayaraman[2], Srinivasa Reddy K.[3]

Submitted: 17/03/2024 **Revised**: 28/04/2024 **Accepted**: 05/05/2024

Abstract: Image forgery detection is a crucial task in digital forensics, aiming to identify manipulated or tampered images. Traditional approaches commonly use handcrafted features, which struggle to effectively capture complex patterns of forgeries. Deep learning techniques have shown promise in this domain, but leveraging hierarchical features while dynamically attending to relevant information remains a challenge. The Attention-aware Hierarchical-feature Fusion Module (AHFM) is proposed to address this issue. The module uses attention mechanisms to selectively fuse hierarchical features extracted from different levels of a convolutional neural network (CNN), enhancing the discriminative power of the network for forgery detection tasks. Experimental results show the AHFM achieves state-of-the-art performance in accuracy and robustness against various forgery techniques. Qualitative analyses provide insights into the interpretability and efficacy of the attention mechanism in identifying forged regions within images. We conduct experiments on a typical dataset of CASIA v1.0 and v2.0 fabricated images, both before and after processing, to illustrate the theoretical concept of the suggested method. Furthermore, we relate the results to those of contemporary methods in order to demonstrate our superior detection rates.

Index Terms: *Image forgery detection, Attention-aware Hierarchical-feature Fusion Module, Deep learning techniques.*

1. Introduction

Digital image editing can alter the semantic content of original images, making them too realistic to distinguish authenticity. Illegal use of these edited images poses a threat to society's stability. Image forgery localization, a forensic task aiming to locate forged regions in investigated images, has gained attention in research and industry. The rapid growth of digital images, enabled by devices like smartphones and tablets, has significantly impacted our modern society [1]. In addition, the advent of affordable, user-friendly image manipulation software [2] has greatly simplified the process of altering such images. In particular, some of these images are tampered in such a way that it is impossible to detect even by humans [3]. Digital watermarking and signatures are distinguishable as active security measures. A manipulated image cannot yield specific details. Watermarking a security structure into images is one way to actively detect tampering; however, the majority of image processing products do not include a watermarking or signature module [4]. Message authentication codes, image checksums, image hashes, and image shielding are just a few of the newer approaches to making images trustworthy and safe, similar to watermarking. Passive forensics is a challenging task in image processing methods [5].There is no one-size-fits-all method to identify a distinctive fake; instead, there are various methods to consider. Passive tempering detection streams use various statistics and semantics of the image's content to examine raw images and pinpoint areas of manipulation [6]. The inherent weaknesses of digital images intensify due to the challenge of determining authenticity solely through human perception. We use either active or passive forgery detection methods to identify these faults and image discrepancies. Most digital images store themselves in the JPEG format. JPEG compression splits the image into 8x8-pixel pieces that do not overlap. A conventional quantization matrix assesses and quantizes the discrete cosine transform (DCT) for each block. Any manipulation of the image will disrupt the local statistics of the block discrete cosine transform (BDCT) coefficients, as the BDCT is the central operation in JPEG compression. To detect counterfeiting, record any changes in the local statistics of the block DCT coefficients.

Another type of image manipulation is known as copy-move forging. This entails taking a small section of one image and superimposing it onto another image in order to make it appear like two separate ones or to hide certain items. Despite being a prevalent image-altering technique, copy-move forgeries may be hard to spot with the human eye if executed meticulously. However, when performing post-processing procedures on the altered images, the methods displayed excessive computational complexity and failed to accurately identify the manipulated regions. On the other

[1*]*Research Scholar, Department of Information Technology, Faculty of Engineering and Technology, Annamalai University Mail: bpravallika03@gmail.com*
[2]*Department of Information technology, Annamalai university, Annamalai nagar, Tamilnadu Mail: sasikala_07@rediffmail.com*
[3]*School of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh, India*
Mail: srinivasareddy.k@vitap.ac.in
***Corresponding Author:** B.Pravallika*
**Research Scholar, Department of Information Technology, Faculty of Engineering and Technology, Annamalai University*

hand, they provide erroneous classification results, are laborious, and are susceptible to subjectivity [7]. After that, researchers came up with image-content-based classification algorithms [8], which get around the problem of subjectivity and have great classification performance. It is standard practice to extract image features using image classification algorithms that rely on low-level visual cues. These approaches excel in image classification in terms of accuracy and temporal complexity. Low accuracy in classification and poor time performance when applied to huge data are two major issues with these approaches' single-feature descriptions and techniques with a single-node design [9].

Based on the above ideas, the AHFM represents a significant advancement in image forgery detection, offering a flexible and interpretable framework for automatically identifying manipulated images. By effectively leveraging hierarchical features and attention mechanisms, the AHFM holds promise for improving the reliability and accuracy of forgery detection systems in real-world applications.

2. Related Work

The literature presents various methods to combat image forgery, with traditional techniques primarily based on specific artifacts. Recently, techniques utilizing CNNs and deep learning have been introduced, highlighting the importance of understanding and addressing these issues in the field.

In [10], the authors presented the default fully connected layers in favour of a pre-trained deep model with a global average pooling (GAP) layer. The GAP layer makes the feature maps and classes effectively dependent on each other. They used heatmap activation, a visual approach, to identify forgeries in images. This technique enables the identification of potentially forged parts of the image. The next step is to choose the most qualified individual and pinpoint where the forgery is. They test the suggested technique using the CoMoFod and MICC datasets. The suggested approach performed well in the extended trials.

In [11], the authors proposed a new YOLO-Forgery model for object insertion forgery detection in images. They used the RGB-to-YUV conversion to enhance the input pictures. The next step in feature extraction using the Clifford gradient approach is to feed in the pre-processed images. In order to detect the area of image fraud with its surrounding boxes, the combined features are input into the YOLOv4 network, which is based on deep learning. They successfully implement the proposed YOLO-Forgery, achieving an average classification accuracy of 99.41%.

In [12] the authors presented a method for fast and accurate counterfeit detection by using techniques such as optimum classification, feature extraction, and pre-processing. The first step involves transforming the image from RGB to L*a*b colour space. They used Eigenvalue Asymmetry (EAS) to identify the KPs and then retrieve texture properties like skewness and haralick. The deep belief network (DBN), a DL classifier (ACSA), uses the adaptive crow search algorithm as the next step. The suggested DBN-ACSA classifier determines whether the detected image is authentic or fraudulent. They experimentally obtained accuracy, precision, recall, F1 score, and ROC values on the Vision, UCID, and CoMoFoD datasets.

In [13], the authors presented a "copy-move forgery detection" system that can still identify forgery areas, even when the it experiences attacks. The process employs the SIFT method to extract keypoints from the fake image, to find areas that are similar to each other that might be counterfeit. The last step in locating copy-move forgery areas is to conduct a region expansion around the found regions. Using a publicly available CoMoFoD dataset, the method outperformed the best "non-deep learning copy-move forging" methods.

In [14], the authors used a CNN-T GAN, a generative adversarial network based on convolutional neural networks, to identify the origin and destination regions of a copy-move-forged image. A generator, similar to the ground truth mask, was created and a discriminator was trained to distinguish between authentic and false image combinations. When the discriminator produced inaccurate results, they turned to the generator to get the final localization mappings for copy-move forgeries. To identify copy-move forgeries, transformers and convolutional neural networks (CNNs) extract local features and global representations. To improve localization performance, especially in source areas, they used a new Pearson correlation layer to align similarity characteristics in the source and target regions. Finding copy-move forgeries has never been easier than in this pioneering work, which employs a transformer for feature extraction. Compared to state-of-the-art approaches, the suggested approach is superior at detecting copy-move areas and differentiating between source and destination regions.

In [15], researchers proposed a novel deep learning-based method for copy-move forgery detection (CMFD). Using CNNs and CovLSTMs is crucial to the suggested approach. This approach first uses a series of convolutions (CNVs), convolutional long short-term memory (ConvLSTM) layers, and pooling layers to extract image features. It then uses feature matching and copy-move forgery detection. Four publicly accessible databases—MICC-F220, MICC-F2000, MICC-F600, and SATs-130—had this approach applied to them. Additionally, they pooled datasets to conduct generalisation tests and address the over-fitting issue. We have also included the results of using the ConvLSTM model alone to illustrate how the performance of the hybrid ConvLSTM and CNN models differs from that of the CNN

model alone. When they set the number of epochs to 100, the suggested technique achieves a high accuracy of 100% with a testing time (TT) of approximately 1 second for certain datasets [21].

3. Proposed Modelling

In this study, Figure 1 is an overview of our framework. We denote the input image as $X \in R^{H\times W\times 3}$, where H and W are the height and width of the image. We first extract the RGB and noise feature from the input image with BayarConv and Transformer Encoder. We suggest using an AHFM to merge hierarchical features from two different domains. To generate the predicted mask, we build a transformer decoder for feature reconstruction.

Dataset: The "Institute of Automation, Chinese Academy of Sciences", generated the gradually interesting and realistic dataset, CASIA v1.0 and v2.0 [16], which we use to assess the suggested method. All 1721 images in CASIA v1.0 are in JPEG format and have not been post-processed. Out of them, 800 are real, and 921 are altered colour images with dimensions of 384×256. However, CASIA v2.0 uses a variety of edge-based post-processing on images of varying sizes. With sizes ranging from 240×160 to 900×600 pixels, CASIA v2.0 includes 7491 genuine and 5123 fake colour images. We may choose between many quality levels for the JPEG and uncompressed versions of the images.

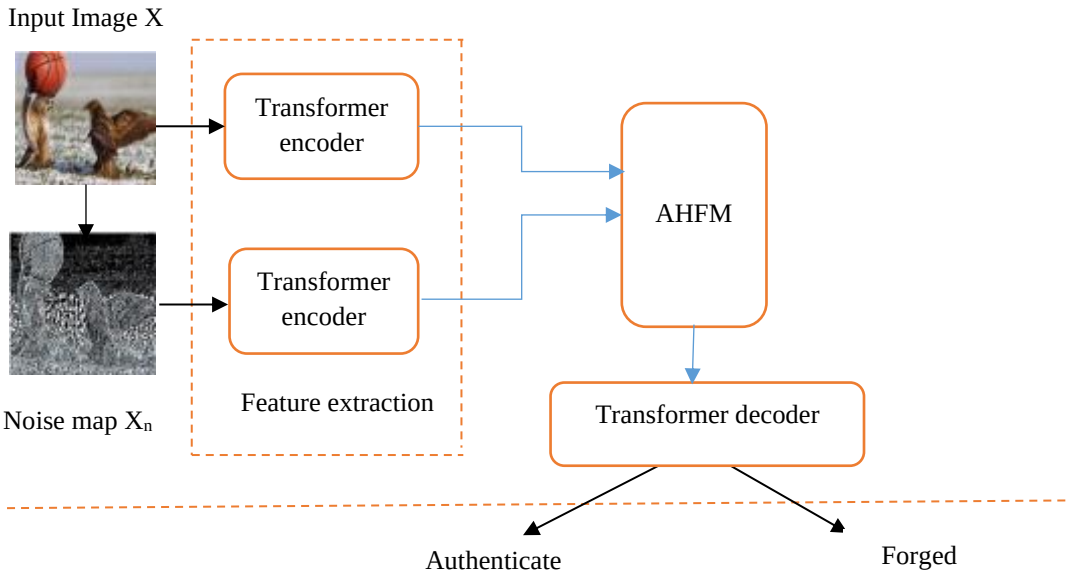

Fig 1. Design framework of the proposed model

Feature extraction

A BayarConv first processes the input image X to extract the noise feature $X_n \epsilon R^{H\times W\times 3}$. Then the input image and noise map are sent to Transformer Encoder. Specifically, we divide X and X_n into patches with size P, and the patch is reshaped to embeddings $X_p \epsilon R^{N\times D}$, where $N = {HW}/{P^2}$ is the number of patches, and D is the dimension of the embedding. Learnable position embeddings $pos \in R^{N\times D}$ are added to image embeddings to produce the sequenced tokens $Z = X_p + pos$, then these tokens are processed through L Transformer layers. The same settlement mentioned above is also performed on the noise branch.

Each patch is then flattened into a vector and linearly projected into an embedding space of dimension D:

$$z_{i,RGB} = W_e.flatten(X_{i,RGB}) + b_e \quad (1)$$

For Noise input:

$$z_{i,Noise} = W_e.flatten(X_{i,\ Noise}) + b_e \quad (2)$$

The Transformer layer consists of a Multi-Head Self-Attention (MSA) block. Self-attention allows each patch to attend to all other patches. For multi-head self-attention, we use h heads.

The input embeddings E are projected into queries Q, keys K, and values V:

$$Q = EW_Q, K = EW_K, V = EW_V \quad (3)$$

Where $W_Q W_K, W_V \epsilon R^{D\times D/h}$ are learnable projection matrices.

The attention scores are computed as

$$Attention(\mathrm{Q},\mathrm{K},\mathrm{V}) = \mathrm{softmax}\frac{QK^T}{\sqrt{d_i}}V \quad (4)$$

Where $d_k = {D}/{h}$ is the dimension of each head.

Multi-head attention combines the outputs from all heads:

$$\mathrm{MultiHead}(\mathrm{Q},\mathrm{K},\mathrm{V}) = Concat(head_1, \dots, head_h)W_O \quad (5)$$

Where each head $head_i = Attention(Q_i, K_i, V_i)$ and $W_O \epsilon R^{D \times D}$ is the output projection matrix.

Figure 2 shows the framework of a dual-input task. Nevertheless, the initial attention mechanism is unable to differentiate between the location data of various input feature sequences. When processing inputs, the original technique by itself takes absolute location into account and appends absolute positional encodings. As follows, the technique considers the shift in focus from a source feature φ to a target feature θ:

$$A_\varphi(\theta) = MultiHead\big(\theta + P_\varphi, \theta + P_\varphi, \varphi\big) \quad (6)$$

where the spatial positional encodings of features θ and φ are represented by P_θ and P_φ, respectively. We use a sine function to create the spatial positional encoding. Equation (5) can be used as an augmentation of attention in one direction or as a co-attention mechanism that takes both directions into account.

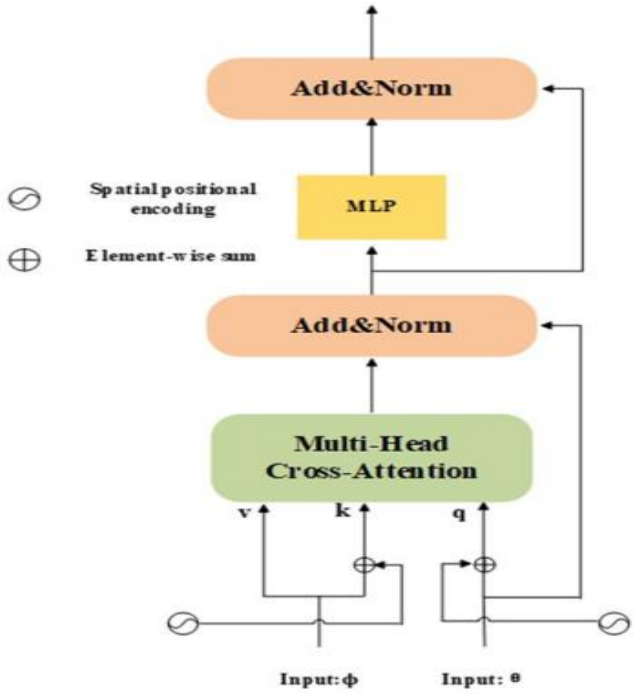

Fig 2. Trabsformer encoder

An additional specific situation is the focus on the feature itself, or self-attention.

$$A_\theta(\theta) = MultiHead(\theta + P_\theta, \theta + P_\theta, \theta) \quad (7)$$

It is not possible to Equations (6) and (7) handle the proposed transformer blocks individually, as shown in Figure 2. Thus, the two modules may be used either in a sequential or parallel fashion. To further improve the model's fitting capabilities, a MLP module is included. A fully connected network with two linear projections and an activation function between them indicated as a Gaussian error linear unit (GELU) is the MLP module.

$$MLP(\theta') = FC_2\Big(GELU\big(FC_1(\theta')\big)\Big) \quad (8)$$

Attention-aware Hierarchical-feature Fusion Module AHFM:

The framework takes as input the raw image data or preprocessed feature representations obtained from a convolutional neural network (CNN) backbone. The feature maps of two branches differ due to their extraction from diverse fields. When looking at data from many domains, a feature fusion module is essential, and a well-designed decoder helps in mask reconstruction. Figure 3 shows that an AHFM may effectively combine hierarchical features from different domains.

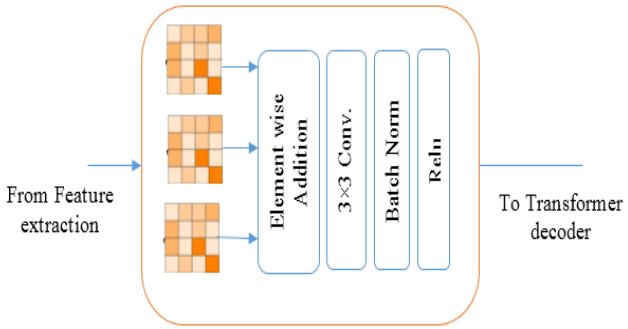

Fig 3. Simplified form of AHFM

Let X_i denote the input feature maps from different levels, where i indexes the levels. We start by computing attention scores for each level.

$$A_i = Attention(X_i) \qquad (9)$$

The attention mechanism Attention(·) computes attention scores based on the input feature maps. This mechanism can vary based on the specific implementation, but let's denote it as a function Attention (·) for now. Once we have the attention scores, we apply a softmax activation to obtain attention weights, ensuring that the weights sum up to 1.

The softmax function ensures that the attention weights α_i are normalized and lie in the range [0, 1], representing the importance of each feature map.

$$\alpha_i = softmax(A_i) \qquad (10)$$

With the attention weights obtained, we combine the input features using these weights.

$$\sum F = \sum_i \alpha_i . A_i \qquad (11)$$

Here, F represents the final fused feature map.

Now, let's examine into more detail for the attention mechanism:

Transformer Decoder

The decoder is designed to learn feature representations of authentic and forged classes, using two learnable class embeddings to effectively segment images. The decoder part of the Transformer architecture can be adapted for this binary classification task. However, since the decoder's primary purpose is to generate output sequences autoregressively. The decoder can be modified to generate a single representation vector for binary classification tasks, using a single-layer decoder with a self-attention mechanism. This mechanism takes encoded input features as key and value, and a learnable query vector representing class information. The decoder can attend to different parts of input features based on class information. A classification layer can predict the probability of input belonging to each class. The entire model is trained end-to-end using a binary cross-entropy loss function, with encoder, decoder, and classification layer parameters optimized jointly to minimize classification loss.

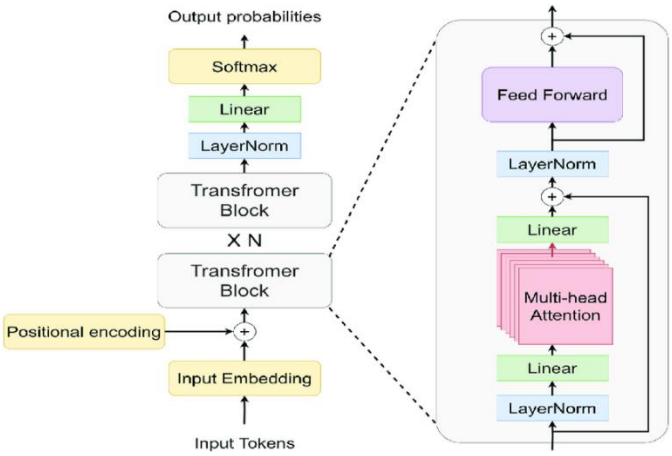

Fig 4. Trabsformer decoder

Let $Y \in R^{L \times D}$ be the input to the decoder, where L is the sequence length and D is the embedding dimension. Positional encoding P_i is added to the input embeddings to retain positional information:

4. Results And Discussions

We use a computer with 4 GB of RAM and an I-3 core CPU running at 2.4 GHz for the various tests. This study's experimental data comes from publicly available, free datasets that include images from CASIA versions 1.0 and 2.0. Python and Anaconda 3.6 are the only development environments used. By plotting the true-positive rate (TPR) and false-positive rate (FPR) on the receiver operating characteristic (ROC) curve, we can see how well the suggested system can classify data. With a low false-positive rate (FPR) and a high true-positive rate (TPR), the classification ability is good, as seen by the curve's proximity to the upper left corner [20]. A few of the most popular ways to evaluate performance are as follows:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FN + FP)} \quad (12)$$

$$Precision = \frac{TP}{(TP + FN)} \quad 13)$$

$$Recall = \frac{TN}{(TN + FP)} \quad (14)$$

$$F1\ score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (15)$$

This outlines the classification of forged images into TP, FN, TN, and FP, with TP representing the number of forged images correctly identified, FN representing the number of forged images correctly identified as authentic, and TN representing the number of authentic images incorrectly identified. The experiment evaluates test images with varying altered regions sizes for both types of forgeries. We test the suggested technique against pre-processing processes such as rotation, deformation, and resizing to determine its resilience. We generate image sets by comparing the modified regions to the target image using the Casia v2.0 and v1.0 datasets.

Authentic image

Forged image

Fig 5: Tested results of CASIA v1.0

Authentic image

Forged image

Fig 6: Tested results of CASIA v2.0

Table 1: Investigational assessment on the CASIA v1.0 Dataset for DL models

ML model	Accuracy	Precision	Recall	F1-score
Baseline CNN	0.9617	0.9778	0.9706	0.9742
Proposed work	0.9891	0.9945	0.9945	0.9945

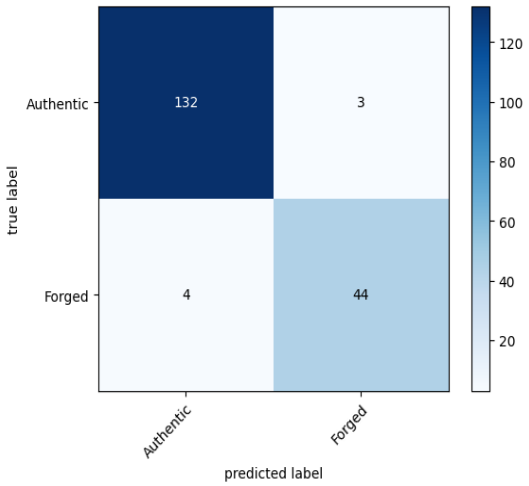

Fig 7: Confusion matrix of Baseline CNN model on CASIA v1.0

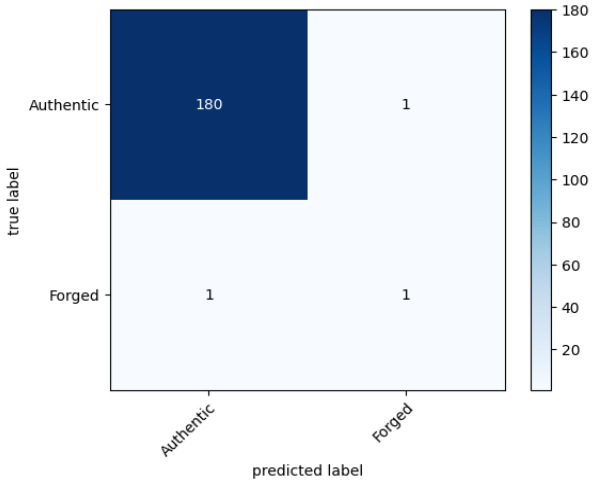

Fig 8: Confusion matrix of proposed model on CASIA v1.0

The proposed work achieved a good accuracy of 0.9891 on the CASIA v1.0 dataset with feature normalization, precision of 0.9945, recall of 0.9945, and an F1-score of 0.9945, but CNN underperformed in all four parameters (see

Figure 9). In contrast to CNN, AHFM uses a linear normalisation algorithm to scale feature values to an expected range, making it more robust against outliers. The model is designed to prevent overfitting and is user-friendly.

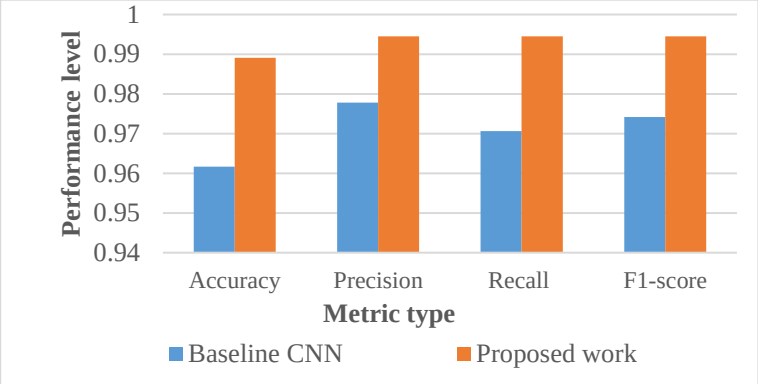

Fig 9: Performance evaluation of ML models on CASIA v1.0 Dataset

Table 2: Investigational assessment on the CASIA v2.0 Dataset for DL models

ML model	Accuracy	Precision	Recall	F1-score
Baseline CNN	0.9790	0.9857	0.9913	0.9885
Proposed work	0.9869	0.9944	0.9914	0.9929

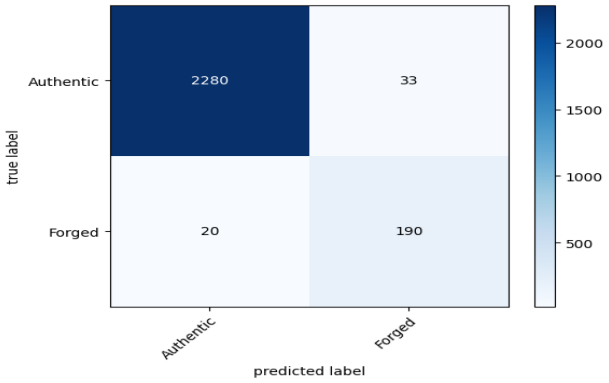

Fig 10: Confusion matrix of Baseline CNN model on CASIA v12.0

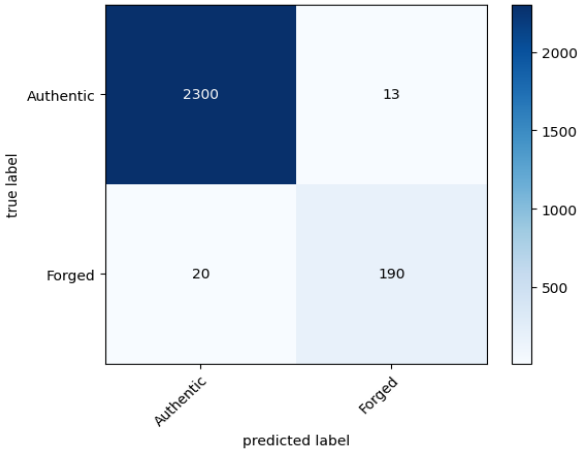

Fig 11: Confusion matrix of proposed model on CASIA v2.0

With feature normalization, the proposed work achieved good accuracy on the CASIA v2.0 dataset of 0.9869, precision of 0.9944, recall of 0.9914, and an F1-score of 0.9929, but CNN underperformed in all four parameters (see Figure 11). Contrary to CNN, AHFM uses a linear normalisation algorithm to scale feature values to a certain range, making it more robust against outliers. Consequently, the model is both simple to use and secure against overfitting. As a result, the model is safe from being overfit and is simple to use.

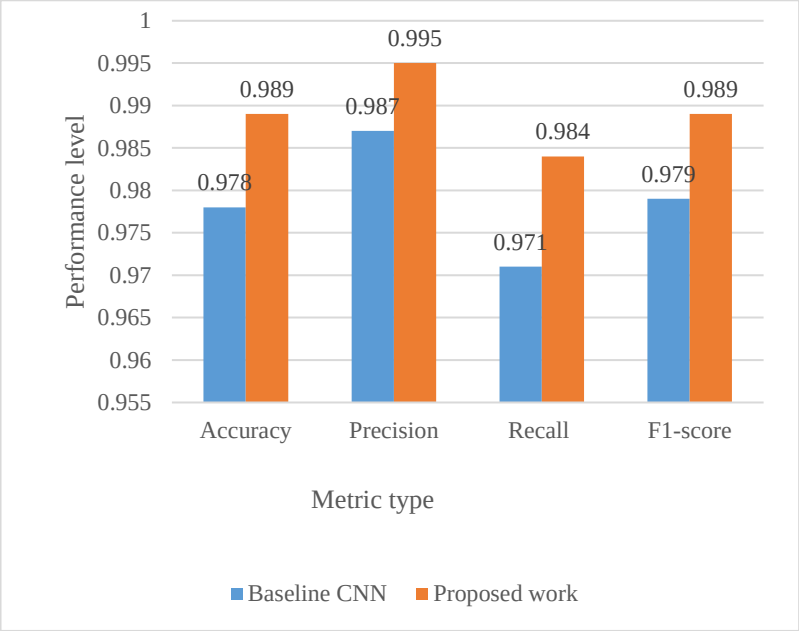

Fig 12: Performance evaluation of ML models on CASIA v2.0 Dataset

Table 3: Comparing the detection accuracy and Execution time of two methods

Method	Total time for execution(sec)	Detection Accuracy (%)
Baseline CNN	55.75	98.14
Proposed work	22.21	99.10

Table 3 displays the results of the two methods for the CASIA v1.0 and v2.0 datasets. As compared to the baseline CNN method, the results show that AHFM (proposed work) uses less features. Additionally, the process of prediction requires less time, with the best execution time of 22.21 seconds as shown in Figure 13.

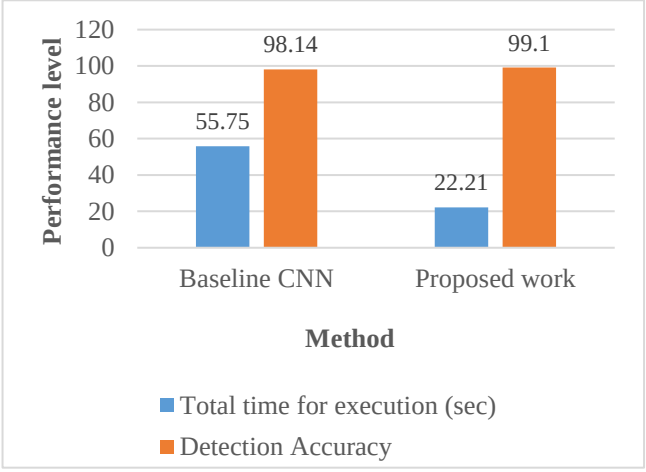

Fig 13: Results of two methods for detection accuracy and execution time

5. Conclusion

In this paper an AHFM for image forgery detection tasks is proposed. The AHFM enhances the discriminative power of convolutional neural networks by effectively leveraging hierarchical features and dynamically attending to relevant information. Experiments on benchmark datasets show the module achieves state-of-the-art performance in accuracy and robustness against various forgery techniques. Qualitative analyses show the interpretability and efficiency of the attention mechanism, enabling high precision identification of forged regions. Further research and development include exploring more sophisticated attention mechanisms, investigating novel fusion strategies, and extending the AHFM to handle multimodal data. Experiments on larger and more diverse datasets could provide further insights into the generalization and robustness of the proposed approach. It has been tested and found to have above-average detection rates of 98.2% for the CASIA v1.0 dataset and 98.5% for the CASIA v2.0 dataset when tested with the standard CASIA dataset for both types of forgeries.

References

[1] Bunk, Jason & Bappy, Jawadul & Mohammed, Tajuddin Manhar & Nataraj, Lakshmanan & Flenner, Arjuna & Manjunath, B. & Chandrasekaran, Shivkumar & Roy-Chowdhury, Amit & Peterson, Lawrence. (2017). Detection and Localization of Image Forgeries using Resampling Features and Deep Learning.

[2] Narasimha, Vadthe, and M. Dhanalakshmi. "Risk Factor of Diabetes with Comorbidity Using Machine Learning Techniques." *ICCCE 2021: Proceedings of the 4th International Conference on Communications and Cyber Physical Engineering*. Singapore: Springer Nature Singapore, 2022.

[3] Muhammad, Ghulam & Al-Hammadi, Muneer & Hussain, Muhammad & Bebis, George. (2014). Image Forgery Detection Using Steerable Pyramid Transform and Local Binary Pattern. Machine Vision and Applications. 25. 985-995. 10.1007/s00138-013-0547-4.

[4] Sabeena, M & Abraham, Lizy & Varghese, Anju. (2021). Digital Image Forgery Detection Using Local Binary Pattern (LBP) and Harlick Transform with classification. 1-6. 10.1109/IPRECON52453. 2021. 9640976.

[5] C. Chen, J. Ni, and J. Huang, "Blind detection of median filtering in digital images: A difference domain-based approach," IEEE Transactions on Image Processing, vol. 22, pp. 4699-4710, 2013.

[6] X. Lin, J.-H. Li, S.-L. Wang, A.-W.-C. Liew, F. Cheng, and X.-S. Huang, "Recent Advances in Passive Digital Image Security Forensics: A Brief Review," Engineering, 2018/02/17/ 2018.

[7] M. D. Ansari, S. P. Ghrera, and V. Tyagi, "Pixel-based image forgery detection: A review," IETE journal of education, vol. 55, pp. 40-46, 2014.

[8] Du PJ, Xia JS, Zhang W, Tan K, Liu Y, Liu SC. Multiple classifier system for remote sensing image classification: A review. Sensors 2012; 12: 4764–4792. https://doi.org/10.3390/s120404764 PMID: 22666057

[9] Zhu S, Sun X, Jin D. Multi-view semi-supervised learning for image classification. Neurocomputing 2016; 208: 136–142.

[10] Fernando B, Fromont E, Tuytelaars T. Mining mid-level features for image classification. Int J Comput Vis. 2014; 108: 186–203.

[11] Mehrjardi, Fatemeh & Latif, Alimohammad & Zarchi, Mohsen. (2023). Copy-Move Forgery Detection and Localization Using Deep Learning. International Journal of Pattern Recognition and Artificial Intelligence. 37. 10.1142/S0218001423520122.

[12] Koshy, Litty & PraylaShyry, S.. (2024). YOLO-FORGERY: Forgery Detection in images via Deep Clifford gradient-based YOLOv4 Network. Signal, Image and Video Processing. 1-9. 10.1007/s11760-023-02854-y.

[13] Archana, M. & Biradar, Deepak & Dayanand, J.. (2023). Image forgery detection in forensic science using optimization based deep learning models. Multimedia Tools and Applications. 1-22. 10.1007/s11042-023-17316-3.

[14] Kalluvilayil Venugopalan, Akash & Gopakumar, Gopalakrishna Pillai. (2023). Keypoint-Based Detection and Region Growing-Based Localization of Copy-Move Forgery in Digital Images. 513-524. 10.1007/978-981-19-7867-8_41.

[15] Zhang, Yulan & Zhu, Guopu & Wang, Xing & Luo, Xiangyang & Zhou, Yicong & Zhang, Hongli & Wu, Ligang. (2022). CNN-Transformer Based Generative Adversarial Network for Copy-Move Source/Target Distinguishment. IEEE Transactions on Circuits and Systems for Video Technology. PP. 1-1. 10.1109/TCSVT.2022.3220630.

[16] Elaskily, Mohamed & Alkinani, Haider & Sedik, Ahmed & Dessouky, Mohamed. (2021). Deep learning based algorithm (ConvLSTM) for Copy Move Forgery Detection. Journal of Intelligent & Fuzzy Systems. 40. 4385-4405. 10.3233/JIFS-201192..

[17] Dong, Jing & Wang, Wei & Tan, Tieniu. (2013). CASIA Image Tampering Detection Evaluation Database. 422-426. 10.1109/ChinaSIP.2013.6625374.

[18] Yuan GL, Xue MG, Han YS, Zhou PC. Mean shift object tracking based on adaptive multi-features fusion, J Comput Res Develop. 2010; 47: 1663–1671.

[19] Wu Y, Li L. Sample normalization methods in quantitative metabolomics. J Chromatography A. 2016; 1430: 80–95.

[20] Narasimha, Vadthe, and Dr M. Dhanalakshmi. "Detection and severity identification of Covid-19 in chest X-ray images using deep learning." *International Journal of Electrical and Electronics Research* 10.2 (2022): 364-369.

[21] Narasimha, Vadthe, and M. Dhanalakshmi. "Identification and Characterization of Effects on Levels of COVID-19 and Diabetes Types using Machine Learning Approaches.".