AUTOMATING IMAGE CAPTIONING WITH AN IMAGE AUTHENTICITY VERIFIER

Thesis Submitted
in Partial Fulfillment of the Requirements for the
Degree of

MASTER OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE

by

NIDHI VARDHAN
(2K22/AFI/13)

Under the Supervision of

Dr. R.K YADAV
Assistant Professor

Department of Computer Science and Engineering

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

May, 2024

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGMENT

I wish to express my sincerest gratitude to **Dr. R.K Yadav** for his continuous guidance and mentorship that he provided during research work. He showed me the path to achieving targets by explaining all the tasks to be done and explained to me the importance of this work as well as its industrial relevance. He was always ready to help me and clear our doubts regarding any hurdles in this project. Without his constant support and motivation, this work would not have been successful.

Place: Delhi **NIDHI VARDHAN**

Date:

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, **Nidhi Vardhan 2K22/AFI/13**, of **M.Tech. (AI)**, hereby certify that the work which is being presented in the thesis entitled "**Automating Image Captioning with an Image Authenticity Verifier**" in partial fulfillment of the requirement for the award of the degree of Master of Technology in Artificial Intelligence, submitted from the Department of Computer Science and Engineering, Delhi Technological University is an authentic record of my own work carried out during the period from to under the supervision of Dr. R.K Yadav.
The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other institute.

Candidate's Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Signature of Supervisor **Signature of External Examiner**

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE BY THE SUPERVISOR

Certified that **Nidhi Vardhan (2K22/AFI/13)** has carried out their research work presented in this thesis entitled "**Automating Image Captioning with an Image Autheticity Verifier**" for the award of **Master of Technology** in Artificial Intelligence from the Department of Computer Science and Engineering, Delhi Technological University, Delhi under my supervision. The thesis embodies results of original work, and studies are carried out by the student herself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

(Dr.R.K YADAV)

(Assistant Professor)
(Department of Computer Science and Engineering)
(Delhi Technological University)

Date:

ABSTRACT

Recent visual analysis and interpretation breakthroughs are largely due to the convergence of artificial intelligence (AI) and computer vision. Deep learning techniques have become a very effective among these methods, in particular for identifying altered photos and generating accurate captions. While the individual modules have made significant progress, there is still much to learn about integrating an image captioner and an image validator into a single framework. Since proper description of visuals is the primary means of understanding visual material, this integrated approach is crucial for the visually handicapped. This type of system can provide an efficient defense against the dissemination of fake or altered photos through simultaneous tempered detection, boosting the dependability and trustworthiness.

In this project, we provide a novel deep learning-based method that combines picture captioning and image verification. This integration produces accurate and efficient subtitles and helps determine the legitimacy of the image, regardless of whether it is tempered or not. The importance of this integration cannot be overstated for visually impaired users, it means that they may now receive accurate descriptions of the visuals in addition to being able to believe that the pictures are legitimate. By protecting against misleading information, this integrated model enhances the user's ability to engage with and understand visual content securely and efficiently.

Several standard datasets are employed to assess the system. The outcomes demonstrate notable enhancements, in the reliability of the verification process and the quality of the descriptions. Based on results incorporating an image validator substantially reduces errors offering a more trustworthy solution for applications in digital asset management, assistive technology and automated content creation. This study addresses challenges in describing images. Makes a significant contribution to artificial intelligence, by introducing a dual component framework. The approach minimizes. Enhances the dependability of generated image descriptions through the use of an image verifier.

LIST OF PUBLICATIONS

- “Comprehensive Study on Automated Image Captioner with an Image Verifier”accepted to be published in ”2nd International Conference on Optimization Techniques in Engineering and Technology” to be held on June 14-15, 2024 at Dronacharya Group of Institutions, Greater Noida , UP, India.

- ”Automated Image Captioner with an Image Verifier” accepted to be published in ”International Conference on Intelligent Computing and Communication Techniques” to be held on 28 & 29 June 2024 at JNU New Delhi, India.

TABLE OF CONTENTS

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Overview

The widespread availability of picture manipulation software has led to increased doubts over the truthfulness of digital image. It can be hard to distinguish the difference between authentic photos and deepfakes and other modified images because they might appear incredibly genuine. Suppose an image of a happy actor and a world leader shaking hands. The description of the image is ”a delighted actor shaking hands with a world leader” which might be true. The verifier can pick up on small changes in skin tone or lighting and detect tampering. This is where the innovative strategy of combining picture captioning with image verification comes out. These effective methods can be combined to provide accurate captions and identify whether an image is altered. It would be beneficial for those who are visually impaired and rely on image descriptions to understand what's in the picture.

Combining two technologies computer vision and deep learning has enhanced the understanding and interpretation of pictures. As a result, sophisticated methods have been developed that enable advanced machines to understand and analyze visual information. These techniques also enable machines to identify whether an image has been altered or not. Various state-of-the-art deep learning models are used in image captioning tasks as well as in image verifiers.

Convolutional Neural Networks have emerged as a key component of image feature extraction models. CNNs have proven to be very effective in detecting image tampering. Pre-trained CNN architectures, VGG/ResNet, are used to extract important features, making the process of creating captions and verifying images easier. CNNs can also learn hierarchical representations directly from raw pixel data. Logical word sequences are produced by recurrent neural networks (RNNs), particularly Long Short-Term Memory (LSTM) networks. RNNs use the visual data that CNN has collected to create word-by-word details that capture the semantic context of the image. In image captioning, encoder-decoder architectures are a famous paradigm where CNNs work as encoders to extract image features and RNNs act as decoders to translate the information extracted from images into comprehensible captions.

Traditionally, sequential models such as RNNs or CNN are used for image captioning tasks. However new approaches to approaching this problem more cohesively and efficiently have emerged with pre-trained language models, such as BERT (Bidirectional Encoder Representations from Transformers) and GPT-2 (Generative Pre-trained Transformer 2). For Image captioning Task, Attention Mechanisms are also very popular. They allow models to dynamically focus on important regions of the image during caption generation. Techniques like Local Binary Patterns (LBP) and Scale-Invariant Feature Transform (SIFT) are used for image tampering detection. They depend on extracting specific features from images, such as noise patterns, lighting inconsistencies, and color histogram discrepancies.

Image synthesis has undergone a revolution due to Generative Adversarial Networks (GANs), which generate extremely realistic and believable artificial images. the misuse of GAN technology to create false information, such as deepfakes or fake news, has raised issues regarding possible effects on society. It isn't easy to distinguish GAN-generated images because of their great visual quality and realism. Since saturation cues quantify the brightness of colors in an image, they effectively detect minute disparities created during the GAN synthesis process[1]. To solve this particular problem several detection strategies, including metadata analysis, crowdsourcing, deep learning models, and traditional photo analysis techniques are used.

1.2 Motivation

The basic motivation for doing this project is to address two challenges: improving accessibility for visually impaired people and countering digital misinformation. Visually challenged people encounter considerable challenges to understanding visual content, which is a common kind of information in the digital age. Existing assistive technologies frequently fail to give detailed and contextually relevant descriptions of images which are also not altered, limiting their independence and involvement in digital worlds. This project aims to provide visually impaired individuals with accurate and detailed image descriptions by developing an integrated system that combines advanced image captioning with robust image verification, thereby improving their ability to understand and interact with visual content independently.

1.3 Problem Statement

As the internet expands progressively more visual, with images playing an increasingly important role in communication, entertainment, and information distribution, the demand for trustworthy and accessible tools to understand and evaluate these images has never been more important. It can be quite difficult for people who are visually impaired to read and understand this visual content. They typically depend on technologies, which translate image information into text/audio. However, current solutions tend to focus on either image captioning or image verification independently.

1.4 Project Objective

This project's primary goal is to develop an advanced system that combines a captioner with an image validator. This system will ensure that photos are properly captioned, verify the authenticity of the images, and remove any inappropriate images from the system in order to protect against digital misinformation. The next project will be very helpful to the visually handicapped since it will allow them to function independently in the digital realm. This model generates contextually-aware and informative captions based on deep learning algorithms, which facilitate viewers' understanding of the images they see. Image verifier combines deep learning algorithms with forensic technologies to determine the authenticity of an image. This involves locating sources, monitoring differences, and providing users with a basis for determining the validity of an image. In this way, when you add these features to the system, it not only fills a need for accessibility to the visually impaired users but also adds to solving a larger social problem of fake news on internet.

1.5 Feasibility study

This project is financially feasible, with an anticipated budget that covers expenses for personnel, technology, and user testing in addition to financial support from grants and partnerships. The project's success is assured through a project management strategy, a team, and continuous user testing. Additionally, compliance with data protection regulations and accessibility guidelines ensures the system's ethicality and security. In summary, the project is well-prepared to enhance accessibility, for the community while also effectively addressing misinformation.

Chapter 2

LITERATURE REVIEW

A significant amount of work has been done on image caption generators as well as on image verifiers. This section explores the model landscapes that are used in each module.

2.1 Image Verifier

Nowadays, tempering an image requires minimal effort. Numerous devices offer user-friendly tools and software that produce such high-quality results that identifying tempering portions is impossible. This allows anyone, even those with limited editing experience, to get high-quality outcomes. Advancements in technology have led to a greater dependence on the Internet for sensitive data transmission. Researchers have increased their focus on digital image forensics due to concerns about authenticity. We have many methods for solving these issues. one method is based on Convolutional Neural Networks[2] that are used to detect manipulated images in medical. Architectures such as region-based and boundary-based models can be used to detect both copy-move and splicing forgeries. Nonetheless, this is not the case with medical images. For privacy reasons, datasets are limited in size, while the images themselves differ significantly from natural ones due to various noise artifacts and anatomical features they possess. This research thus seeks to explore possibilities of using deep learning effectively for detecting copy-move forgery in medical images to cement its position in ensuring that the integrity of this vital data is maintained.

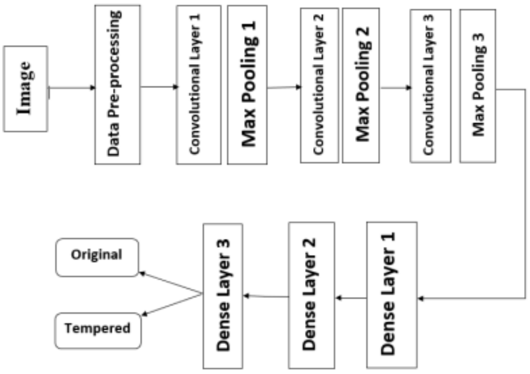


Figure 2.1: Copy move forgery Algorithm.

Another proposed method is based on dual stream-fast RCNN[3].It consists of RGB and noise streams. By determining the consistency of boundary vision in RGB images and detecting noise features, such as the difference between original and changed images. In RGB streams ,VGG19 is used to recognize object edges and identify changing regions. The Noise Stream uses the Spatial Rich Model (SRM) for analyzing noise features which extracted from filtering layers. This SRM eliminates redundancy. Billinear pooling merges RGB and Noise Stream features, conserving crucial information and enhancing detection. The model obtained an accuracy of 93.1% on the Columbia dataset and 83.9% on the CASIA dataset.

Anupam Mishra et al.[4] proposed a CNN with Error Level Analysis technique for image verification tasks. Convolution layers are used to extract features such as texture and edges from images in a Convolutional Neural Network. In the pooling layer, the results of these convolution layers are processed reducing the dimension while maintaining important features like edges, textures, and so on. The results of convolution layers are then processed in a pooling layer, which decreases spatial dimension while preserving significant features. The output of the pooling layers then passes through fully connected layers, which are utilized to forecast picture categorization. Error Level Analysis is used to determine the level of error contained in an image. This technique obtained 87.75 percent accuracy. The CASIA2 dataset is used.

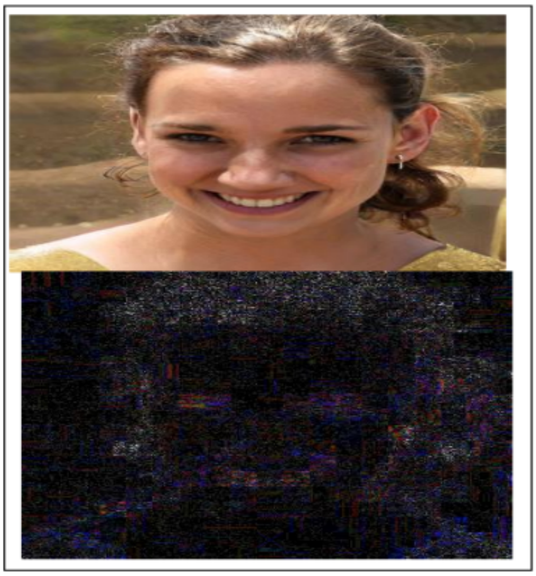


Figure 2.2: Real image to ELA.

Youssef William et al.[5] used two common methods for detecting altered images, which are copy-move and splicing. After feature extraction, they used the match point technique using Scale-Invariant Feature Transform(SIFT) and Speeded-Up Robust Features(SURF). For splicing detection, they extract edges from integral part of images of Y, Cb, and Cr components, applying the Gray-Level Co-occurrence Matrix (GLCM) to form feature vectors. These vectors are then input into a Support Vector Machine (SVM) classifier. SURF's main advantage is the remarkably low computational load, combined with good performance and detection of scaled or rotated objects. Results indicate that SURF feature extraction is more efficient than SIFT for copy-move detection, achieving an 80% accuracy in detecting tampered images. Processing images in the YCbCr color model yields promising results for splicing detection, achieving a 99% true positive rate. In fig:2.3 the block diagram of image tempering detection is shown. CASIA dataset is used.

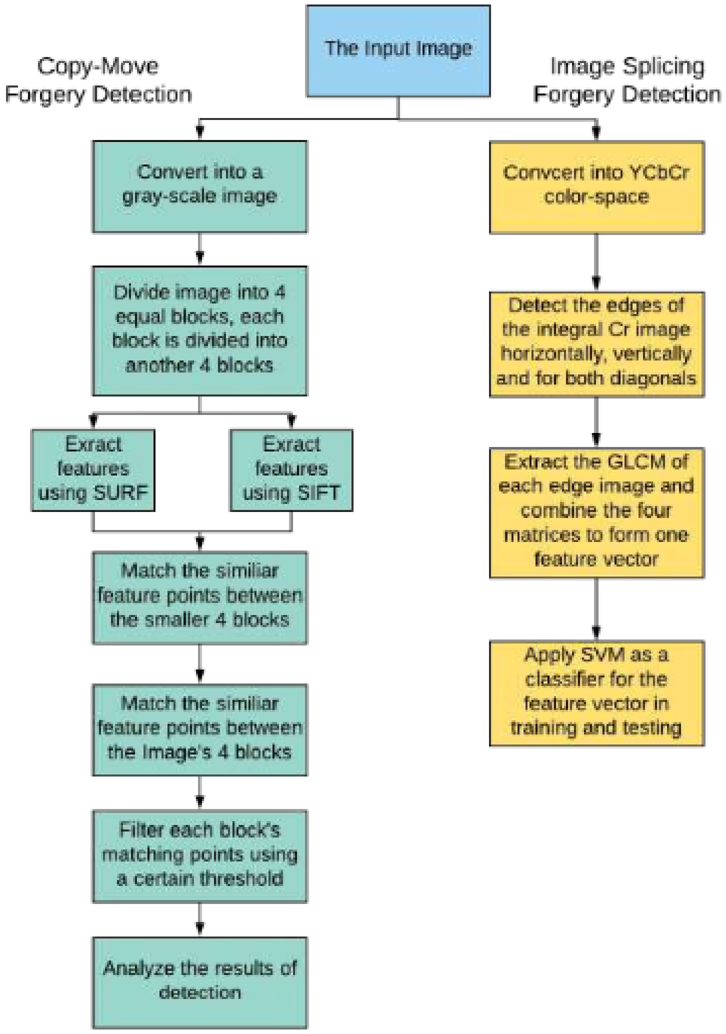

The Input Image
Copy-Move Forgery Detection
Image Splicing Forgery Detection
Convert into a gray-scale image
Divide image into 4 equal blocks, each block is divided into another 4 blocks
Exract features using SURF
Exract features using SIFT
Match the similiar feature points between the smaller 4 blocks
Match the similiar feature points between the Image's 4 blocks
Filter each block's matching points using a certain threshold
Analyze the results of detection
Convcert into YCbCr color-space
Detect the edges of the integral Cr image horizontally, vertically and for both diagonals
Extract the GLCM of each edge image and combine the four matrices to form one feature vector
Apply SVM as a classifier for the feature vector in training and testing

Figure 2.3: Block Diagram of Image tempering detection.

2.2 Image Captioner

A CNN-RNN models are used for image captioning tasks. Specifically, ResNet50/VGG for CNN and Long Short Term Memory(LSTM) for RNN.ResNet50 is used for extracting image features and feeding them into an LSTM for caption generation. ResNet50 prevents degradation and vanishing gradient problems in the neural nets during training and helps in maintaining good accuracy. It has 50 layers and can be optimized. After caption generation, text caption is converted into an audio form using gTTS API. The results show that ResNet50 performed better than VGG16 with an accuracy of 73%.Flickr8K dataset is used. The fig:2.4 shows the architecture of image captioning based on the CNN-RNN model[6].

Another model is based on Adaptive attention[7].The inception-V3 model used to extract various global spatial features and Adaptive Attention helps to decide whether to use those image features or not. This model improves on a CNN model by adding batch normalization to stabilize and accelerate learning, factorized convolutions for more effective operations, an auxiliary classifier to give the network more guidance during training and label smoothing for better training. The attention mechanism dynamically focuses on different regions of an image while generating each word in the

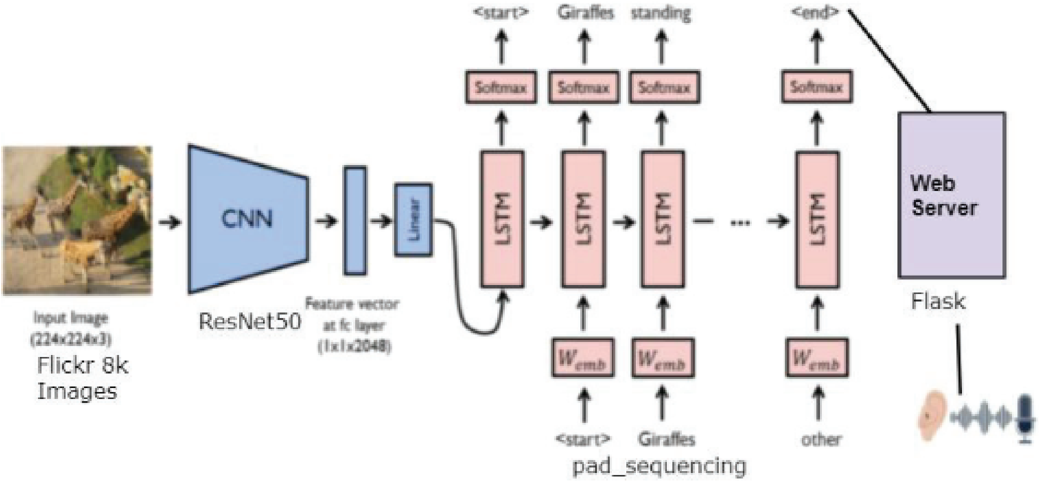


Figure 2.4: Image Captioning Model System Architecture.

caption using the Bi-LSTM model.Bi-LSTM preserves both past and future information by processing input data in bi-directional. The attention mechanism maps unique features with the corresponding text description, enabling accurate localization of text in the image region. Flickr8k dataset is used. This model gave a 0.712 BLEU-1 score. The fig:2.5 shows the architecture of Inception-V3 with Adaptive Attention Bi-LSTM model.

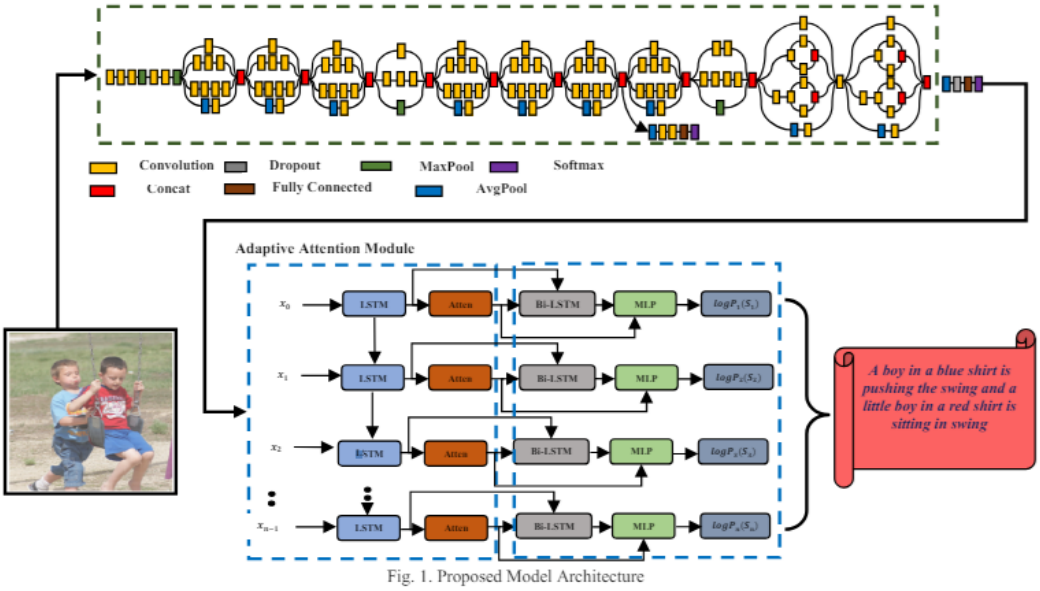


Figure 2.5: Inception-V3 with Adaptive Attention Bi-LSTM model Architecture.

Another method that is proposed is dependent upon both YOLOV5 and BiLSTM combined [8]. YOLOv5 + Bi-LSTM model solves the problem which CNN-LSTM model brings about, hence eliminating Gradient Explosion. YOLOv5 is a fast and precise method of object detection which segments images into grids such that a grid cell contains multiple anchor boxes for predictions. Each anchor box only picks out one object while it also shows how sure it is about its choice. The fig:2.6 illustrates the object detection using YOLO.

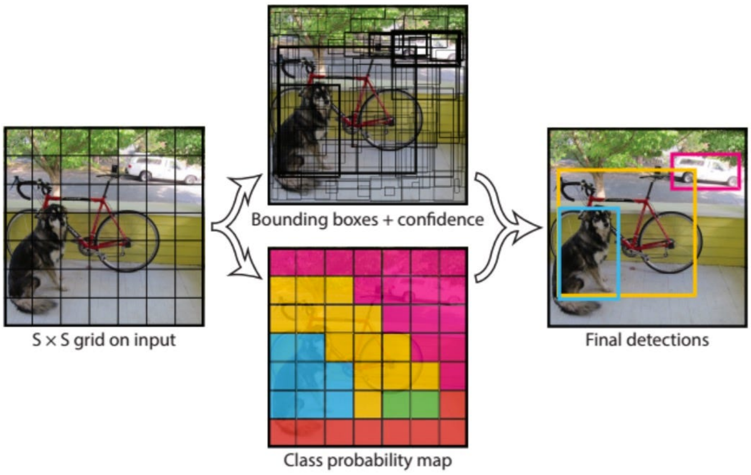


Figure 2.6: Object Detection using YOLO

Bidirectional LSTM(BiLSTM) is used to construct captions after features have been extracted.BilSTM enables input to flow in both directions capturing both past and future information. The Flicker8k dataset is used. This model gave a 0.79 BLEU-1 score. There are five modules in this model as shown in fig:4[8]. Image Pre-processing, Image segmentation, Image feature extraction, Image classification, and Image captioning. Pre-processing enhances the quality of images by removing any discrepancies. Image segmentation is a process of dividing an image into multiple groups based on their pixels. This technique reduces the complexity of the original image. Feature extraction helps in selecting and merging variables into featuresFeature extraction helps in selecting and merging variables into features.This method helps in reducing the amount of data. Image classification categorizes an entire image into multiple classes, with each image anticipated to belong to just one class.fig:2.7 illustrates the Flow Diagram of YOLOv5 and BiLSTM Working.

Another efficient model is based on BERT[9] for image captioning task. For the CNN, the Xception model is used. The last layer is removed to obtain output from a penultimate layer. From the input image of dimensions 299x299x3, extract features resulting in an image feature vector has 2048 values. For RNN, three different models BERT(Birectional Encoder Representation from Transformer), LSTM & GRU(Gated Recurrent Units) are used.BERT is used to decode features of images and generate the textual description.It reads text bi-directionally to capture more information. They use an attention mechanism to collect information about the context of words and encode it into vectors.BERT uses a multihead-attention model which enables to process of multiple positions in the input and enhances its ability to capture complex context within the text.LSTM is designed to eliminate the problem of long-term dependencies in sequential data. They predict the next word based on previous words and also decide which information is to store in their memory, ensuring that important information is retained for future predictions.GRU is the simple version of LSTM with similar abilities. They solved long-term dependency problems by using a gating mechanism to control the flow of information through the networks. Computationally efficient than LSTM networks. Dropout regularization is used to prevent overfitting. fig:2.8 Shows the Comparison Table.

Yeonju kim et al[10] solved the problem of dataset bias. When the image captioner

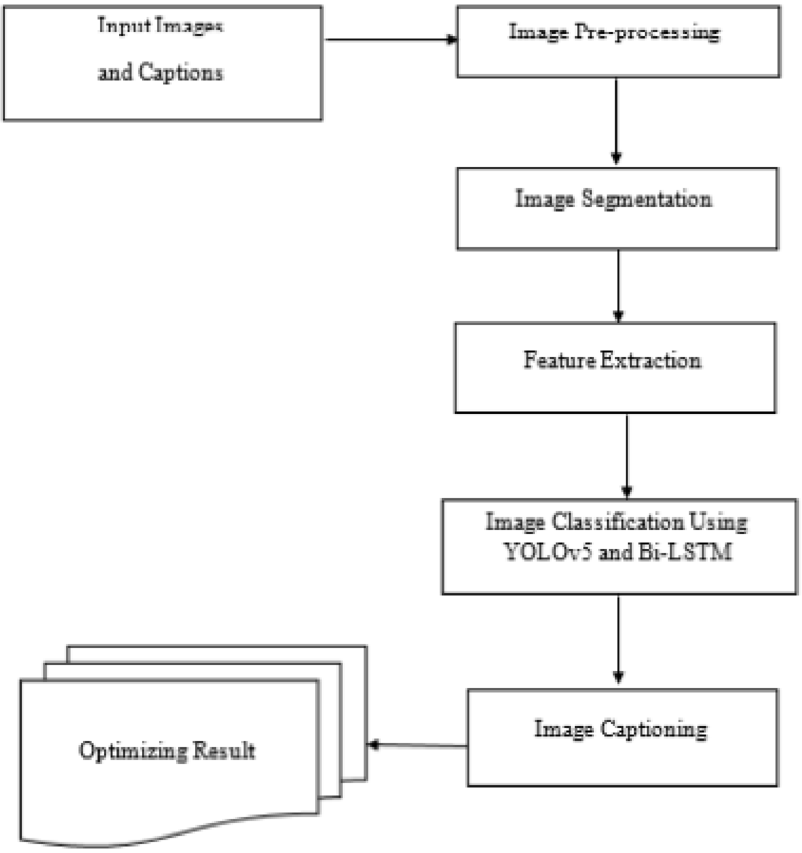
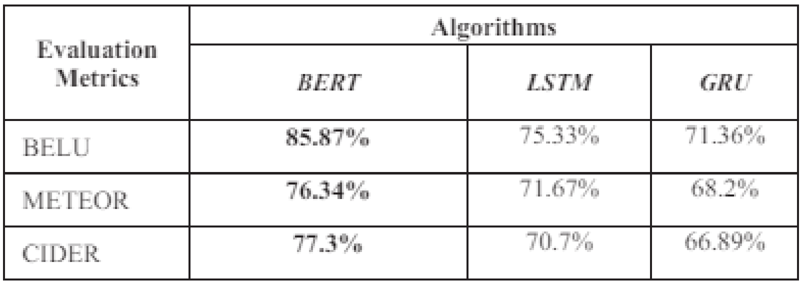


Figure 2.7: Flow Diagram of YOLOv5 and BiLSTM Working



Evaluation Metrics	Algorithms		
	BERT	*LSTM*	*GRU*
BELU	**85.87%**	75.33%	71.36%
METEOR	**76.34%**	71.67%	68.2%
CIDER	**77.3%**	70.7%	66.89%

Figure 2.8: Comparison Table

predicts a word,it should be based on visual evidence not on contextual evidence from the database.Casual graph is used to solved this issue.They proposed the method CLIP Cofounder-Free captioning network based on Casual graph.The key idea is to construct a comprehensive confounder, termed ”global visual confounder” using the pre-trained vision-language model CLIP. CLIP, trained on a large-scale dataset with sentence supervision, provides rich visual features, including objects and predicates, enabling the representation of abundant visual information. By controlling this confounder during training, the model learns to avoid spurious correlations, thereby improving the quality of generated captions.The proposed strategy, which is called C2Cap, uses the backdoor adjustment methodology together with a causal perspective to address the problem of inaccurate correlations in picture captioning.In the causal graph of image captioning, the visual feature X is connected to the context vector M via the true causal effect path $X \rightarrow M$, while the backdoor path $X \leftarrow Z \rightarrow M$ introduces spurious correlations due to the confounder Z. Used backdoor adjustment technique to solve this issue, which involves blocking the path $Z \rightarrow X$.The pre-trained CLIP model is used to generate a large confounder dictionary known as C2Dictionary, which is then used to build the confounder Z.

The two main components of the C2Cap model are the creation of a pre-defined confounder dictionary (C2Dictionary) utilizing CLIP features and the conditioning of a transformer-based image caption model on this dictionary. Training photos are used to extract CLIP features, which are then clustered using k-means and stacked in the C2Dictionary for each cluster centroid.The MSCOCO benchmark yielded a 0.891 BLEU-1 score for this model.

The proposed approach[11] combines the power of Vision Transformer to capture spatial features from individual frames of the input video with the capability of transformer-based language models to understand video contexts, providing a viable option for modeling video understanding and captioning considering both temporal and spatial behaviors. This research intends to examine the efficacy of integrating Vision Transformer and Transformer to generate meaningful, contextually relevant video captions by rigorously comparing at the levels of established tasks and how they compare to baseline methods, e.g., VGG-16 to potentially pushing the boundaries of current capabilities in video understanding.fig:2.9 Shows the combine Model of ViT & Transformer and fig:2.10 shows the Image extraction using Vision Transformer.

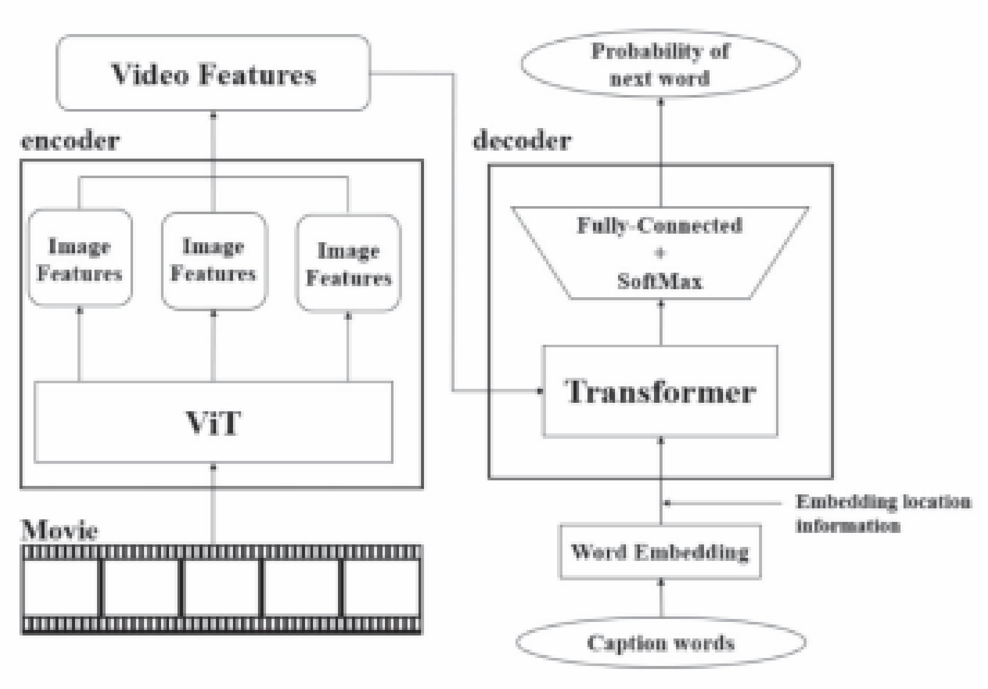


Figure 2.9: Combine Model of ViT & Transformer

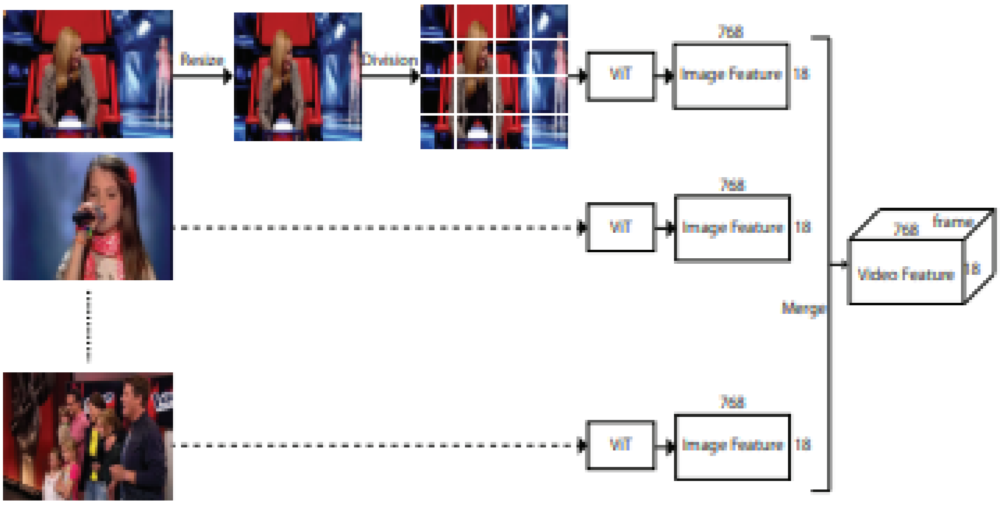


Figure 2.10: Image extraction using Vision Transformer

Chapter 3

METHODOLOGY

The purpose of devising an integrated system that merges image caption generation and image verification together is to come up with a global solution for both meaningful descriptions of pictures and also the authenticity of images. This system has two principal parts that use sophisticated approaches like deep learning for better performance. The first module, which is the image verifier, identifies manipulated photos by employing fine-tuned Convolutional Neural Networks (CNNs) with Error Level Analysis (ELA). After the images have been confirmed as unmodified or original, they are passed onto the second module; this implies that there is no alteration in the image. An image is uploaded to the system by a user. The picture is first preprocessed, which involves resizing, scaling, and normalizing its pixel to standarize the image. The next step is Error Level Analysis (ELA), which detects differences in compression levels and uses this information to create an ELA version of the image that highlights places that may need to be altered. After ELA, Feature extraction is done by a fined tuned Convolutional Neural Network from both the original and ELA photos. The CNN calculates a tampering probability score based on the variables it has examined, indicating the likelihood of picture manipulation. This score is compared to a preset threshold to determine if the image is considered tampered with or real. If the image is found to be authentic, it advances to the next module and if an image is found to be tempered then it remove from the system.The Verified images are then passed through image caption generation. For Caption Generator two models are used. CNN-RNN model and ViT-Bert model. The first model starts by extracting features of verified images using pre-trained convolutional neural networks. These extracted features then serve as input to a Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) model designed for caption generation. And the last step is processing the final caption so as to ensure that it's grammatically correct and consistent. Lastly, the user is provided with the revised caption describing accurately the visual content, along with the verified image. The second model, ViT component specializes in understanding visual content by transforming images into sequences of tokens, while BERT, renowned for its language understanding capabilities, processes these tokenized sequences to generate accurate and contextually relevant captions.

3.1 Image Verifier

3.1.1 Fine-Tuned CNN & ELA

Image verifier Steps :

1. Dataset Collection

- We have merged two datasets: ”CASIA2” and another that contains real and GAN-generated images for image verifier.For image captioner, We have used the ”flickr8k” dataset and this Text Data preprocess the captions, tokenize them, and create a vocabulary. Also, create mappings between words and indices for encoding and decoding.

2. Data Preprocessing:

- It involves several steps which include resizing the image and normalization of pixel values of an image. Augmentation techniques should be applied such as rotations, scaling, flips and color adjustments. This increases the dataset size and variability, thus helping in model generalization.

3. Error Level Analysis (ELA):

- In order to do this, you take the image and save it at a particular compression quality that you know. Then later, when you have to compare it with the original image, you can identify which areas differ by looking at the error levels: they help you see where the inconsistency in compression quality lies across different regions of the images. ELA helps in the identification of inconsistencies thus exposing areas that have been manipulated or edited thereby providing crucial insight.

  Formula for ELA calculation of an image :

  $$\text{ELA} = |\text{original} - \text{recompressed}| \tag{3.1}$$

  where,

  - I_{original} is the pixel value of the original image.
  - $I_{\text{recompressed}}$ is the pixel value of the recompressed image.
  - $|\cdot|$ denotes the absolute difference between the corresponding pixel values.

Figure 3.1: Real Image

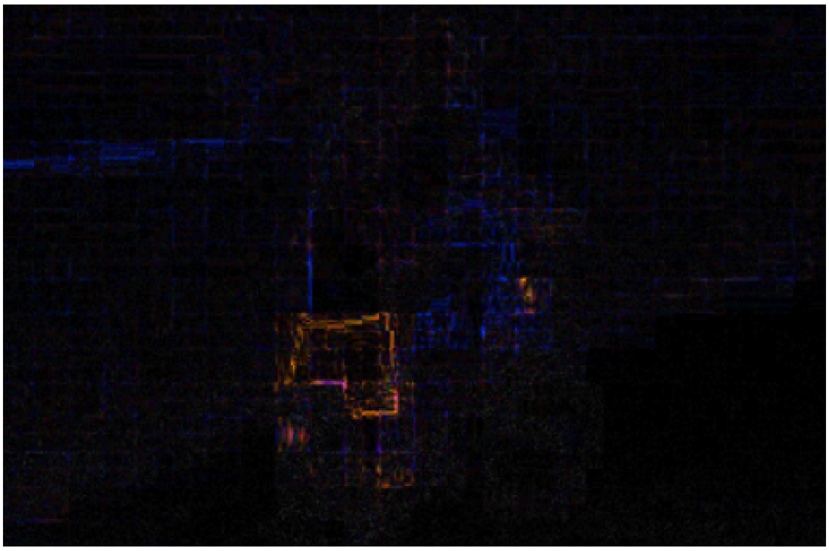


Figure 3.2: ELA of real Image

4. Training & Feature Extraction:

- Model Selection: Select a Vgg16 CNN model that has already been trained .
- Fine-tuning: Fine-tuning: Swap out the pre-trained model's last few classification layers for layers that are appropriate for binary classification (authentic or not).
- Train the Model: When you're training your model, the first step is to split your dataset into test, validation, and training sets. Use the training set to validate the model first; tweak the hyperparameters based on feedback and then train the model.

5. Model Evaluation: Evaluate the model's performance using the Confusion Matrix.

A confusion matrix is a performance measurement for machine learning classification problems where the output can be two or more classes. It is a table with four different combinations of predicted and actual values.

Accuracy: Measures the proportion of correct predictions out of the total predictions.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.2}$$

Precision (Positive Predictive Value): Measures the accuracy of positive predictions.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3.3}$$

3.2 Image Captioner

Verified Images are passed to the image captioner for generating accurate captions.Generation can be done using two methods:

3.2.1 CNN & LSTM Model

Fig: 3.3 shows the flow diagram of image verifer & Image captioner using CNN+LSTM

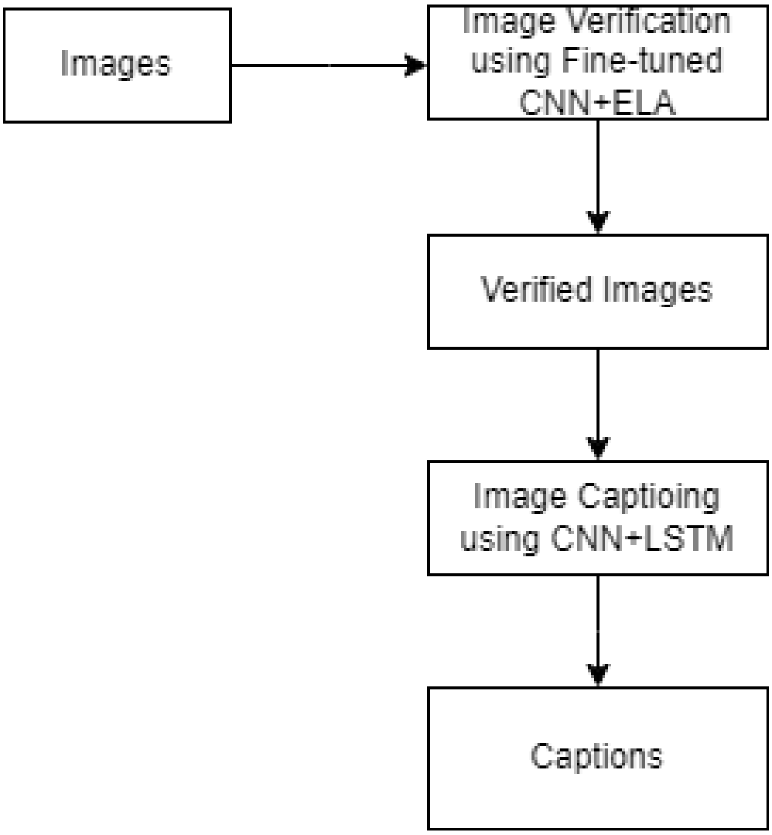

Images
Image Verification using Fine-tuned CNN+ELA
Verified Images
Image Captioing using CNN+LSTM
Captions

Figure 3.3: Flow Diagram

Image Captioner Architecture:

- Image Feature Extraction: The CNN must be used to extract features from the images. A pretrained CNN model, VGG16 can be used to this end. When using the CNN as a feature extractor, the fully connected layers of the network should be removed. In order not to train the CNN during caption generation, freeze the layers of the network.
  Feature Extraction:
  $$f = \text{VGG16}(I) \tag{3.4}$$
  Where:
  - f represents the feature vector extracted from the image I using the VGG16 model.
- LSTM for Caption Generation: Design an LSTM-based sequence model to generate captions. The LSTM takes the image features from the CNN as input and generates a sequence of words as output.
  Caption Generation with LSTM:
  $$h_t = \text{LSTM}(w_{t-1}, h_{t-1}, f) \tag{3.5}$$

Where:

  - h_t is the hidden state of the LSTM at time step t.
  - w_{t-1} is the word generated at time step $t-1$.
  - h_{t-1} is the hidden state of the LSTM at time step $t-1$.
  - f is the feature vector from VGG16.

- Embedding Layer: Map words to high-dimensional vectors using an embedding layer.

- Decoder LSTM: the decoded embedded vectors are used to generate a word-by-word caption.

- Output Layer: Use a softmax layer to predict the next word in the sequence. Word Prediction:

$$P(w_t|w_{t-1}, f) = \text{softmax}(Wh_t + b) \tag{3.6}$$

Where:

  - $P(w_t|w_{t-1}, f)$ is the probability of the word w_t given the previous word w_{t-1} and the image features f.
  - W and b are the weight matrix and bias term for the output layer.

3.2.2 CNN & LSTM model Limitations

- CNN tends to ignore the smallest implication in photos, resulting in overly general headlines.

- For long sequences, it is difficult for LSTM to maintain consistent context, resulting in less logical description.

- This project requires a large number of annotated datasets, which are labor-intensive to create and may not cover all situations.

- Limited contextual understanding: Because models cannot fully understand context, they often generate technically correct but irrelevant headlines.

- Risk of overfitting: This limits generalization, as the model may perform well on training data but perform poorly on new, untested images.

- Bias propagation: This technique may lead to biased or incorrect labels by inheriting bias from the training data.

- Static image representation: In various scenarios, fixed-size representations may not capture all the details required.

3.2.3 ViT & BERT

Fig: 3.4 shows the flow diagram of the image verifer and Image captioner using ViT+BERT Model. Here, Images are passed through an image verifier which verifies the authenticity of an image. After, Verified images are passed to the image captioner which generates a caption using ViT and BERT model.

Vision Transformer:
The Transformer framework was originally designed for natural language processing, but Vision Transformer (ViT) is an innovative model architecture that adapts it to image recognition challenges. To preserve spatial information, the image is first divided

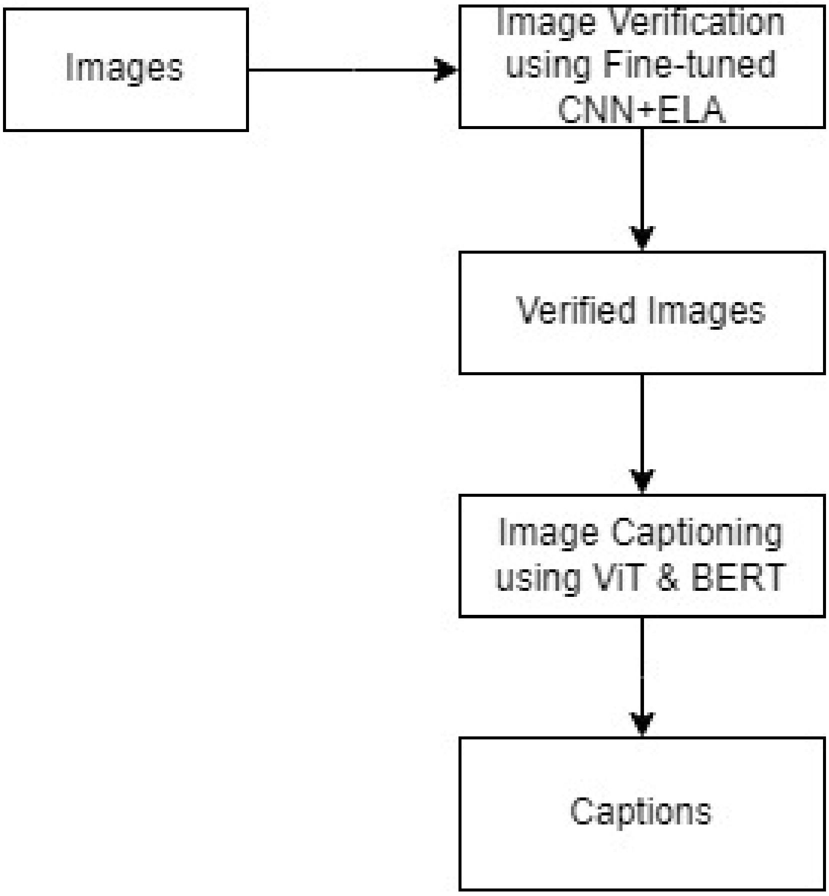

Images
Image Verification using Fine-tuned CNN+ELA
Verified Images
Image Captioning using ViT & BERT
Captions

Figure 3.4: Flow Diagram

into a series of fixed-size blocks. Then, these patches are linearly embedded and enhanced by positional encoding.
The Transformer encoder receives these embedded patches and uses a self-attention mechanism to capture the global context of the entire image. ViT outperforms traditional convolutional neural networks (CNN) on large datasets in capturing long-range dependencies and global features. However, to achieve optimal results, ViT requires extensive computing resources and training data.
Fig:3.5 [12] Shows the Architecture of Vision Transformer.

Feature extraction:

ViT decomposes the image into a series of patches and processes these patches in a similar way to how BERT processes text tokens. Positional embeddings are added to each linearly embedded patch before being sent to the transformer.

Self-attention mechanism:

Compared with traditional CNN, ViT models the relationship between different image components through self-attention and more successfully captures the global context.

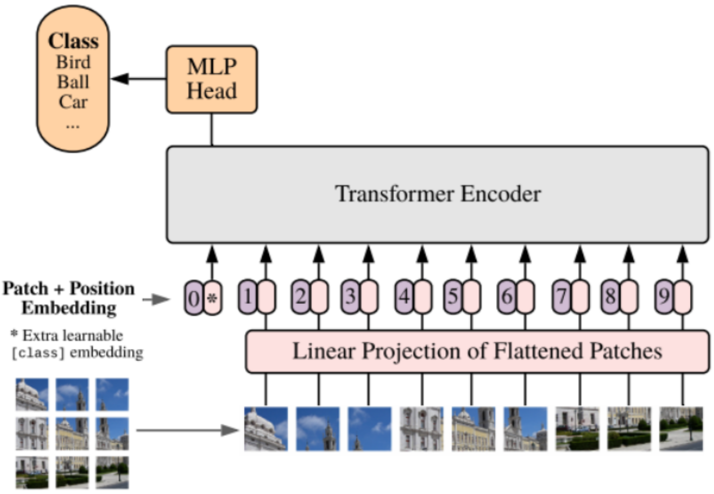

Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0 *
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Figure 3.5: Vision Transformer

BERT: aThe development of language comprehension and generation in machines has undergone a transformational change since the development of BERT (Bidirectional Encoder Representation from Transformers). Unlike previous models, which read text only from left to right, BERT uses a structure called a transformer to read text both from right to left and from left to right to take into account both the left and the right context of every token in a sentence. Fig:3.6 [12] Shows the Architecture of BERT Language Modeling: Text written in natural language is understood and produced by

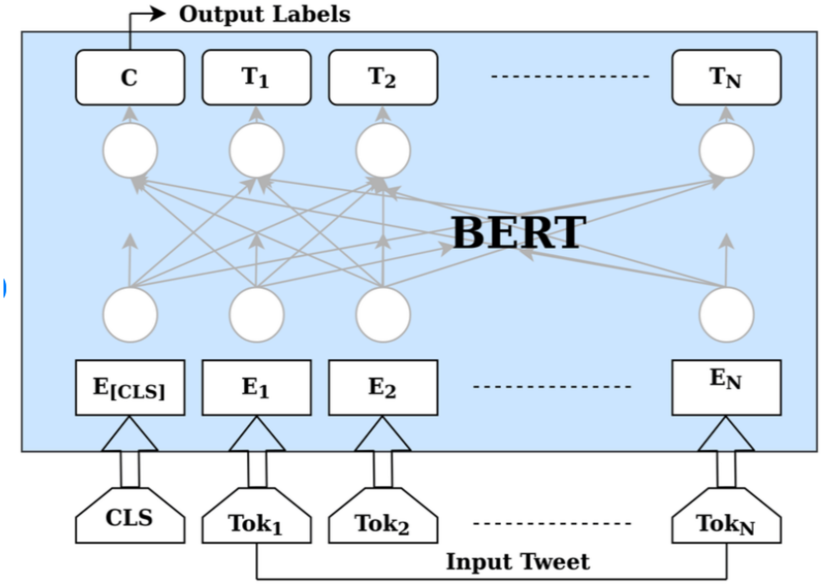


Figure 3.6: BERT Model

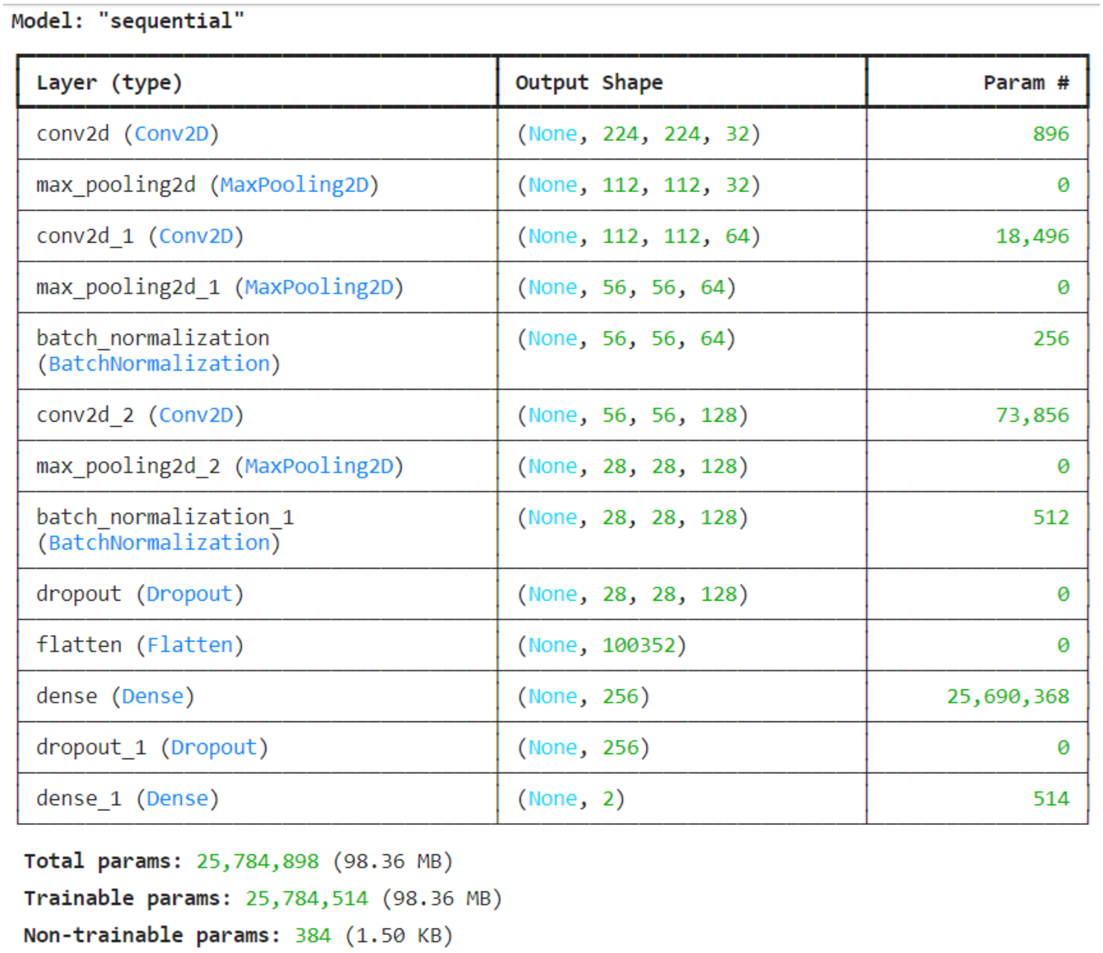
Language Modeling: Written text in natural language can be understood and generated using BERT due to its ability to model the context of a word on both its left and right side by utilizing bidirectional processing of the input text. Caption Generation: ViT produces the visual features that are required to produce a fitting caption. BERT or a transformer model that is very similar to BERT (a Transformer) takes in the visual features as input to generate the corresponding caption.

Chapter 4

RESULTS & ANALYSIS

4.1 Image Verifier

- Image Verifier:The integrated system first verifies the authenticity of uploaded images using techniques like Error Level Analysis (ELA) and a fine-tuned Convolutional Neural Network (CNN).Authentic images are allowed to proceed to the captioning module, ensuring that only reliable visual content is captioned.

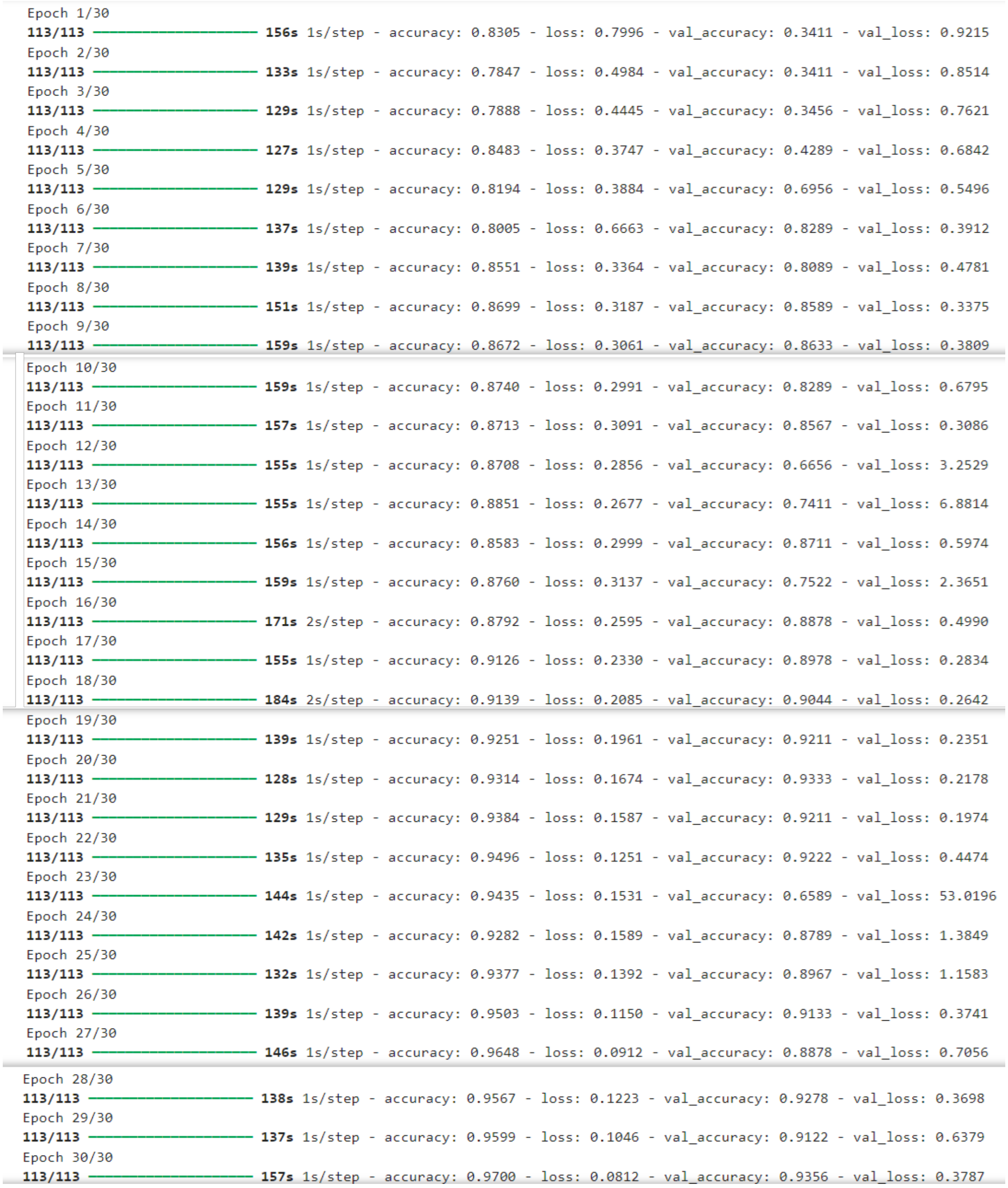


```
Model: "sequential"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━┓
┃ Layer (type)                        ┃ Output Shape               ┃         Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━┩
│ conv2d (Conv2D)                     │ (None, 224, 224, 32)       │             896 │
├─────────────────────────────────────┼────────────────────────────┼─────────────────┤
│ max_pooling2d (MaxPooling2D)        │ (None, 112, 112, 32)       │               0 │
├─────────────────────────────────────┼────────────────────────────┼─────────────────┤
│ conv2d_1 (Conv2D)                   │ (None, 112, 112, 64)       │          18,496 │
├─────────────────────────────────────┼────────────────────────────┼─────────────────┤
│ max_pooling2d_1 (MaxPooling2D)      │ (None, 56, 56, 64)         │               0 │
├─────────────────────────────────────┼────────────────────────────┼─────────────────┤
│ batch_normalization                 │ (None, 56, 56, 64)         │             256 │
│ (BatchNormalization)                │                            │                 │
├─────────────────────────────────────┼────────────────────────────┼─────────────────┤
│ conv2d_2 (Conv2D)                   │ (None, 56, 56, 128)        │          73,856 │
├─────────────────────────────────────┼────────────────────────────┼─────────────────┤
│ max_pooling2d_2 (MaxPooling2D)      │ (None, 28, 28, 128)        │               0 │
├─────────────────────────────────────┼────────────────────────────┼─────────────────┤
│ batch_normalization_1               │ (None, 28, 28, 128)        │             512 │
│ (BatchNormalization)                │                            │                 │
├─────────────────────────────────────┼────────────────────────────┼─────────────────┤
│ dropout (Dropout)                   │ (None, 28, 28, 128)        │               0 │
├─────────────────────────────────────┼────────────────────────────┼─────────────────┤
│ flatten (Flatten)                   │ (None, 100352)             │               0 │
├─────────────────────────────────────┼────────────────────────────┼─────────────────┤
│ dense (Dense)                       │ (None, 256)                │      25,690,368 │
├─────────────────────────────────────┼────────────────────────────┼─────────────────┤
│ dropout_1 (Dropout)                 │ (None, 256)                │               0 │
├─────────────────────────────────────┼────────────────────────────┼─────────────────┤
│ dense_1 (Dense)                     │ (None, 2)                  │             514 │
└─────────────────────────────────────┴────────────────────────────┴─────────────────┘

 Total params: 25,784,898 (98.36 MB)
 Trainable params: 25,784,514 (98.36 MB)
 Non-trainable params: 384 (1.50 KB)
```

Figure 4.1: CNN Model Summary



```
Epoch 1/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 156s 1s/step - accuracy: 0.8305 - loss: 0.7996 - val_accuracy: 0.3411 - val_loss: 0.9215
Epoch 2/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 133s 1s/step - accuracy: 0.7847 - loss: 0.4984 - val_accuracy: 0.3411 - val_loss: 0.8514
Epoch 3/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 129s 1s/step - accuracy: 0.7888 - loss: 0.4445 - val_accuracy: 0.3456 - val_loss: 0.7621
Epoch 4/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 127s 1s/step - accuracy: 0.8483 - loss: 0.3747 - val_accuracy: 0.4289 - val_loss: 0.6842
Epoch 5/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 129s 1s/step - accuracy: 0.8194 - loss: 0.3884 - val_accuracy: 0.6956 - val_loss: 0.5496
Epoch 6/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 137s 1s/step - accuracy: 0.8005 - loss: 0.6663 - val_accuracy: 0.8289 - val_loss: 0.3912
Epoch 7/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 139s 1s/step - accuracy: 0.8551 - loss: 0.3364 - val_accuracy: 0.8089 - val_loss: 0.4781
Epoch 8/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 151s 1s/step - accuracy: 0.8699 - loss: 0.3187 - val_accuracy: 0.8589 - val_loss: 0.3375
Epoch 9/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 159s 1s/step - accuracy: 0.8672 - loss: 0.3061 - val_accuracy: 0.8633 - val_loss: 0.3809
Epoch 10/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 159s 1s/step - accuracy: 0.8740 - loss: 0.2991 - val_accuracy: 0.8289 - val_loss: 0.6795
Epoch 11/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 157s 1s/step - accuracy: 0.8713 - loss: 0.3091 - val_accuracy: 0.8567 - val_loss: 0.3086
Epoch 12/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 155s 1s/step - accuracy: 0.8708 - loss: 0.2856 - val_accuracy: 0.6656 - val_loss: 3.2529
Epoch 13/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 155s 1s/step - accuracy: 0.8851 - loss: 0.2677 - val_accuracy: 0.7411 - val_loss: 6.8814
Epoch 14/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 156s 1s/step - accuracy: 0.8583 - loss: 0.2999 - val_accuracy: 0.8711 - val_loss: 0.5974
Epoch 15/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 159s 1s/step - accuracy: 0.8760 - loss: 0.3137 - val_accuracy: 0.7522 - val_loss: 2.3651
Epoch 16/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 171s 2s/step - accuracy: 0.8792 - loss: 0.2595 - val_accuracy: 0.8878 - val_loss: 0.4990
Epoch 17/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 155s 1s/step - accuracy: 0.9126 - loss: 0.2330 - val_accuracy: 0.8978 - val_loss: 0.2834
Epoch 18/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 184s 2s/step - accuracy: 0.9139 - loss: 0.2085 - val_accuracy: 0.9044 - val_loss: 0.2642
Epoch 19/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 139s 1s/step - accuracy: 0.9251 - loss: 0.1961 - val_accuracy: 0.9211 - val_loss: 0.2351
Epoch 20/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 128s 1s/step - accuracy: 0.9314 - loss: 0.1674 - val_accuracy: 0.9333 - val_loss: 0.2178
Epoch 21/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 129s 1s/step - accuracy: 0.9384 - loss: 0.1587 - val_accuracy: 0.9211 - val_loss: 0.1974
Epoch 22/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 135s 1s/step - accuracy: 0.9496 - loss: 0.1251 - val_accuracy: 0.9222 - val_loss: 0.4474
Epoch 23/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 144s 1s/step - accuracy: 0.9435 - loss: 0.1531 - val_accuracy: 0.6589 - val_loss: 53.0196
Epoch 24/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 142s 1s/step - accuracy: 0.9282 - loss: 0.1589 - val_accuracy: 0.8789 - val_loss: 1.3849
Epoch 25/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 132s 1s/step - accuracy: 0.9377 - loss: 0.1392 - val_accuracy: 0.8967 - val_loss: 1.1583
Epoch 26/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 139s 1s/step - accuracy: 0.9503 - loss: 0.1150 - val_accuracy: 0.9133 - val_loss: 0.3741
Epoch 27/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 146s 1s/step - accuracy: 0.9648 - loss: 0.0912 - val_accuracy: 0.8878 - val_loss: 0.7056
Epoch 28/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 138s 1s/step - accuracy: 0.9567 - loss: 0.1223 - val_accuracy: 0.9278 - val_loss: 0.3698
Epoch 29/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 137s 1s/step - accuracy: 0.9599 - loss: 0.1046 - val_accuracy: 0.9122 - val_loss: 0.6379
Epoch 30/30
113/113 ━━━━━━━━━━━━━━━━━━━━ 157s 1s/step - accuracy: 0.9700 - loss: 0.0812 - val_accuracy: 0.9356 - val_loss: 0.3787
```

Figure 4.2: Epoch-wise Performance Metrics

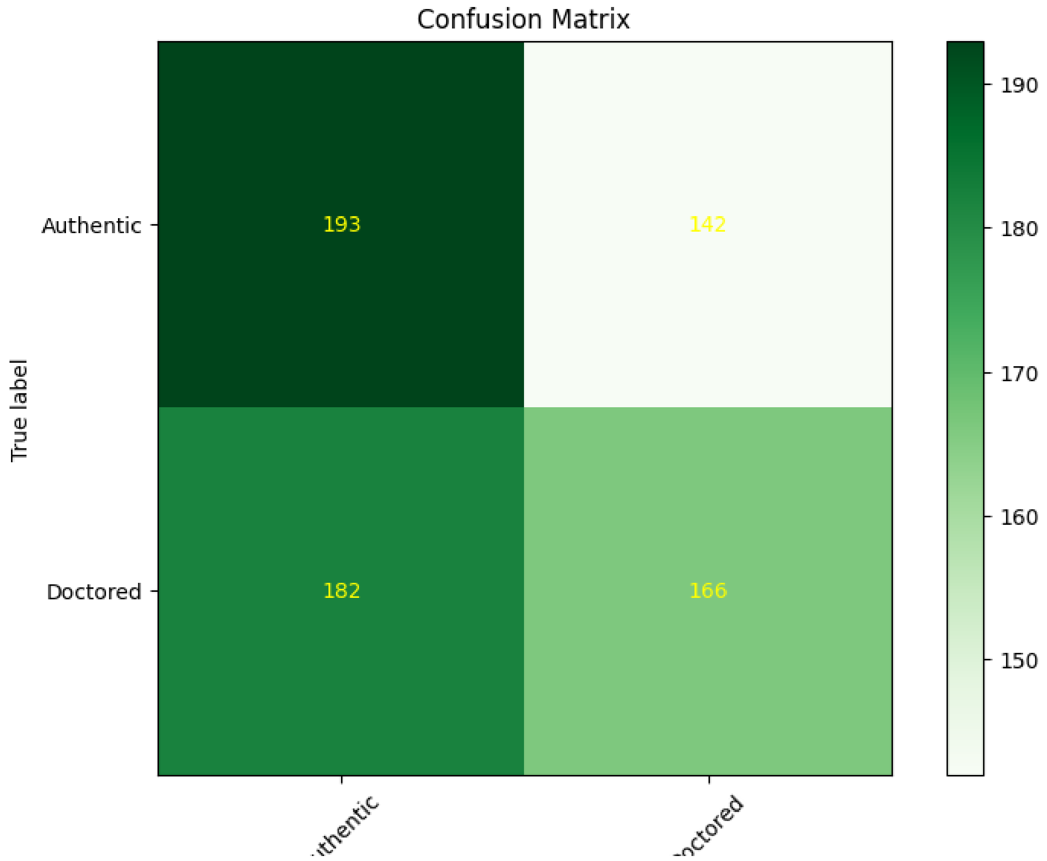
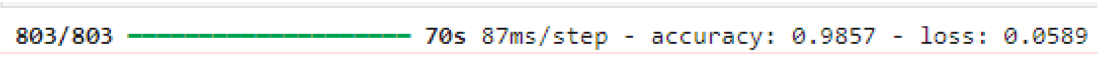
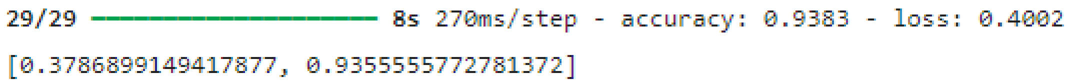


Figure 4.3: Confusion Matrix



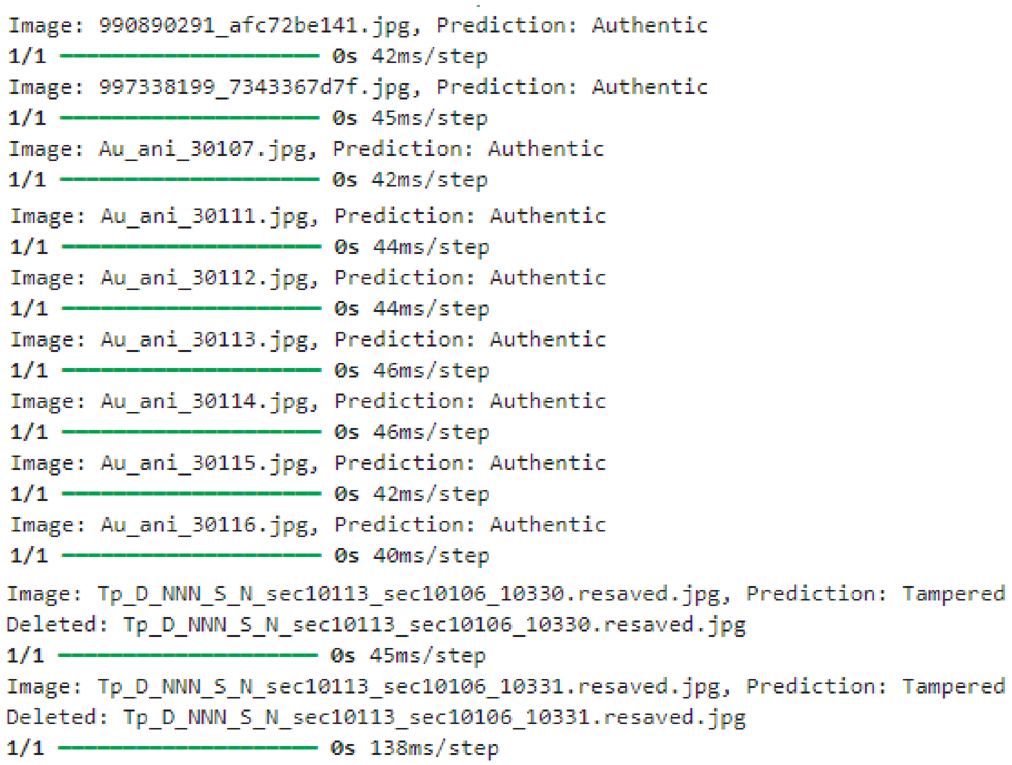
```
803/803 ━━━━━━━━━━━━━━━━━━━━ 70s 87ms/step - accuracy: 0.9857 - loss: 0.0589
```

Figure 4.4: Training loss & Training Accuracy



```
29/29 ━━━━━━━━━━━━━━━━━━━━ 8s 270ms/step - accuracy: 0.9383 - loss: 0.4002
[0.3786899149417877, 0.9355555772781372]
```

Figure 4.5: Validation loss and validation Accuracy



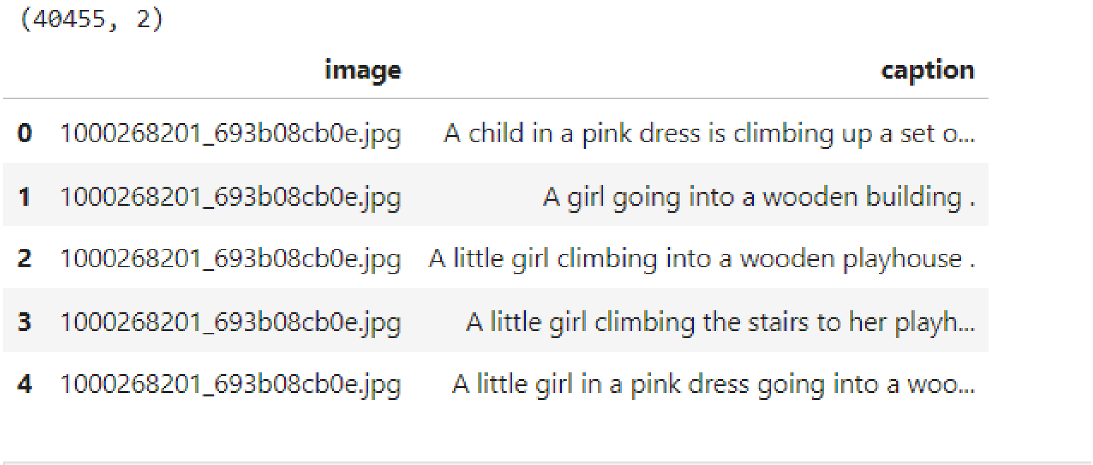
```
Image: 990890291_afc72be141.jpg, Prediction: Authentic
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
Image: 997338199_7343367d7f.jpg, Prediction: Authentic
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
Image: Au_ani_30107.jpg, Prediction: Authentic
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
Image: Au_ani_30111.jpg, Prediction: Authentic
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
Image: Au_ani_30112.jpg, Prediction: Authentic
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
Image: Au_ani_30113.jpg, Prediction: Authentic
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
Image: Au_ani_30114.jpg, Prediction: Authentic
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
Image: Au_ani_30115.jpg, Prediction: Authentic
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
Image: Au_ani_30116.jpg, Prediction: Authentic
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
Image: Tp_D_NNN_S_N_sec10113_sec10106_10330.resaved.jpg, Prediction: Tampered
Deleted: Tp_D_NNN_S_N_sec10113_sec10106_10330.resaved.jpg
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
Image: Tp_D_NNN_S_N_sec10113_sec10106_10331.resaved.jpg, Prediction: Tampered
Deleted: Tp_D_NNN_S_N_sec10113_sec10106_10331.resaved.jpg
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 138ms/step
```

Figure 4.6: Output of Image Verifier

References	Model	Accuracy
Marra F et al.[13]	CNN	81.51%
McCloskey S et al.[1]	LSVM	86%
Mishra A et al.[4]	CNN + ELA	87.75%
Our Model	Fine-Tuned CNN + ELA	93.56%

Table 4.1: Comparison of various Image Verifier approaches.

4.2 Image Captioner



```
(40455, 2)
```

	image	caption
0	1000268201_693b08cb0e.jpg	A child in a pink dress is climbing up a set o...
1	1000268201_693b08cb0e.jpg	A girl going into a wooden building .
2	1000268201_693b08cb0e.jpg	A little girl climbing into a wooden playhouse .
3	1000268201_693b08cb0e.jpg	A little girl climbing the stairs to her playh...
4	1000268201_693b08cb0e.jpg	A little girl in a pink dress going into a woo...

Figure 4.7: Images and their Captions

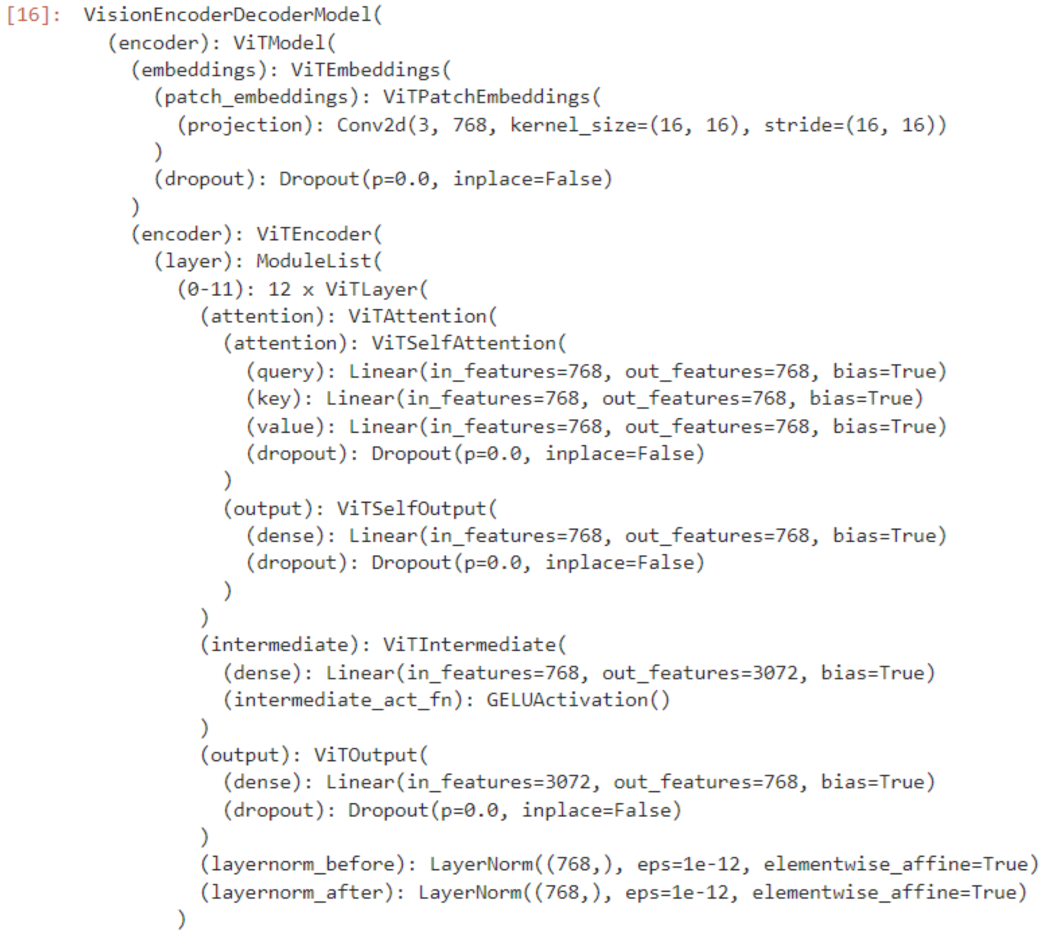
Metric	Score
ROUGE-1	0.356
ROUGE-2	0.288
ROUGE-L	0.358
ROUGE-Lsum	0.358

Table 4.2: ROUGE Scores on Train Set

Metric	Score
ROUGE-1	0.349
ROUGE-2	0.320
ROUGE-L	0.365
ROUGE-Lsum	0.345

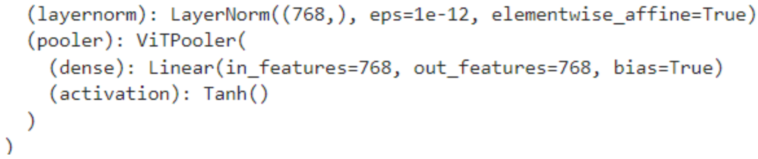
Table 4.3: ROUGE Scores on Validation Set

VisionEncoderDecoderModel:



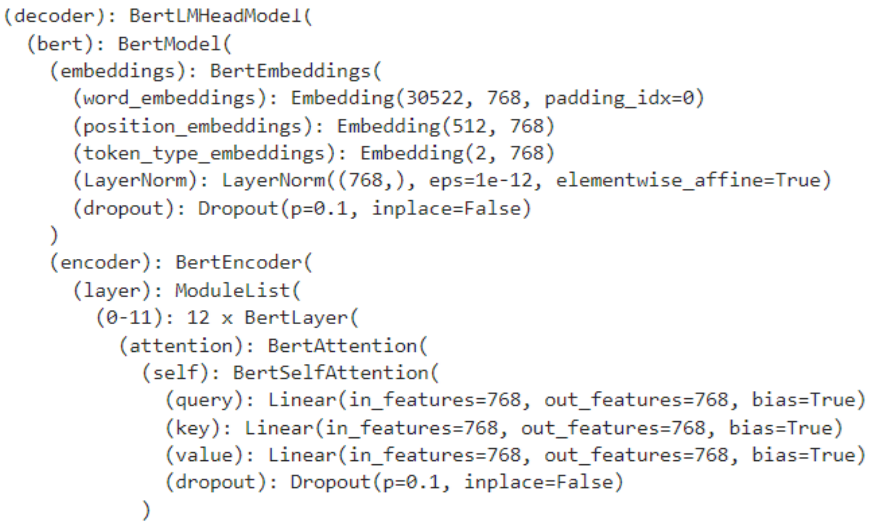
```
[16]:  VisionEncoderDecoderModel(
         (encoder): ViTModel(
           (embeddings): ViTEmbeddings(
             (patch_embeddings): ViTPatchEmbeddings(
               (projection): Conv2d(3, 768, kernel_size=(16, 16), stride=(16, 16))
             )
             (dropout): Dropout(p=0.0, inplace=False)
           )
           (encoder): ViTEncoder(
             (layer): ModuleList(
               (0-11): 12 x ViTLayer(
                 (attention): ViTAttention(
                   (attention): ViTSelfAttention(
                     (query): Linear(in_features=768, out_features=768, bias=True)
                     (key): Linear(in_features=768, out_features=768, bias=True)
                     (value): Linear(in_features=768, out_features=768, bias=True)
                     (dropout): Dropout(p=0.0, inplace=False)
                   )
                   (output): ViTSelfOutput(
                     (dense): Linear(in_features=768, out_features=768, bias=True)
                     (dropout): Dropout(p=0.0, inplace=False)
                   )
                 )
                 (intermediate): ViTIntermediate(
                   (dense): Linear(in_features=768, out_features=3072, bias=True)
                   (intermediate_act_fn): GELUActivation()
                 )
                 (output): ViTOutput(
                   (dense): Linear(in_features=3072, out_features=768, bias=True)
                   (dropout): Dropout(p=0.0, inplace=False)
                 )
                 (layernorm_before): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
                 (layernorm_after): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
               )
```

Figure 4.8: ViTModel(Encoder)



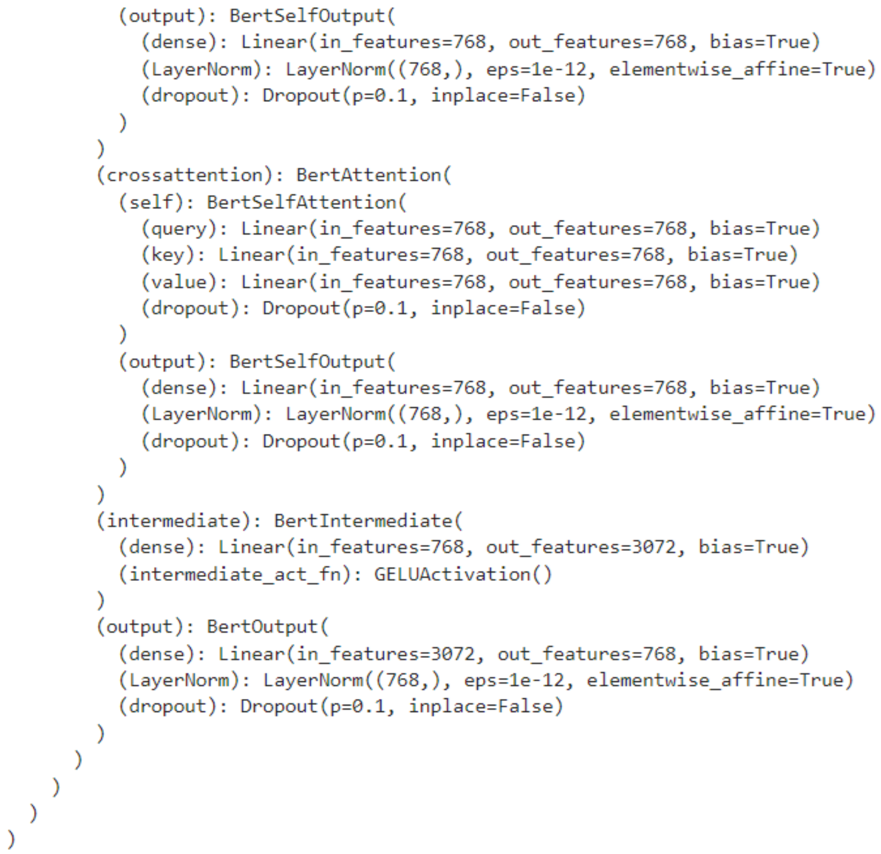
```
    (layernorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
    (pooler): ViTPooler(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (activation): Tanh()
    )
  )
```

Figure 4.9: Normalization Layer



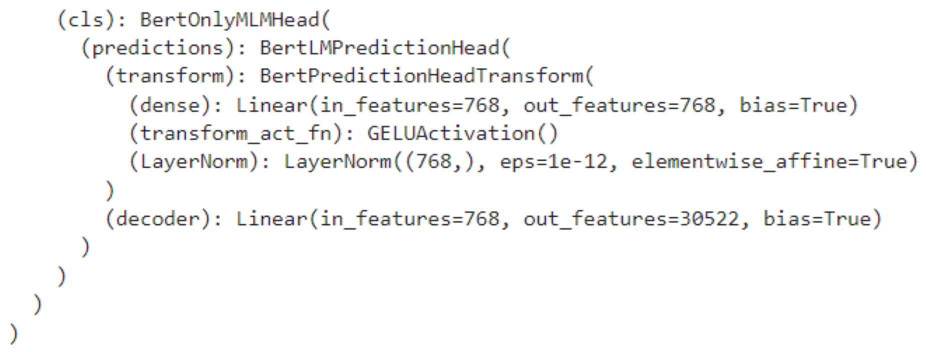
```
(decoder): BertLMHeadModel(
  (bert): BertModel(
    (embeddings): BertEmbeddings(
      (word_embeddings): Embedding(30522, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (token_type_embeddings): Embedding(2, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): BertEncoder(
      (layer): ModuleList(
        (0-11): 12 x BertLayer(
          (attention): BertAttention(
            (self): BertSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
```

Figure 4.10: Decoder: BertLMHeadModel



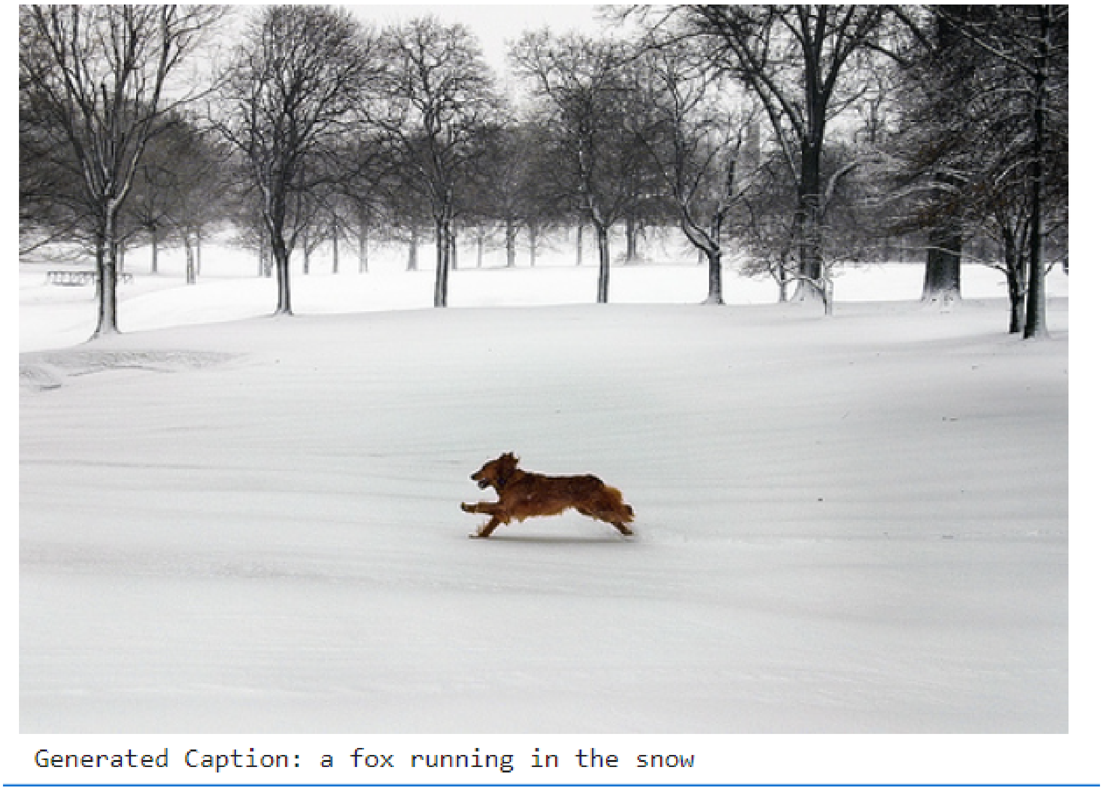
```
          (output): BertSelfOutput(
            (dense): Linear(in_features=768, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
        (crossattention): BertAttention(
          (self): BertSelfAttention(
            (query): Linear(in_features=768, out_features=768, bias=True)
            (key): Linear(in_features=768, out_features=768, bias=True)
            (value): Linear(in_features=768, out_features=768, bias=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
          (output): BertSelfOutput(
            (dense): Linear(in_features=768, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
        (intermediate): BertIntermediate(
          (dense): Linear(in_features=768, out_features=3072, bias=True)
          (intermediate_act_fn): GELUActivation()
        )
        (output): BertOutput(
          (dense): Linear(in_features=3072, out_features=768, bias=True)
          (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (dropout): Dropout(p=0.1, inplace=False)
        )
      )
    )
  )
)
```

Figure 4.11: BERT Layer includes Attention Mechanisms, Intermediate Transformations, and Output Layer



```
    (cls): BertOnlyMLMHead(
      (predictions): BertLMPredictionHead(
        (transform): BertPredictionHeadTransform(
          (dense): Linear(in_features=768, out_features=768, bias=True)
          (transform_act_fn): GELUActivation()
          (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        )
        (decoder): Linear(in_features=768, out_features=30522, bias=True)
      )
    )
  )
)
```

Figure 4.12: Classification Head (BertOnlyMLMHead)



```
Generated Caption: a fox running in the snow
```

Figure 4.13: Output of Image Captioner

Chapter 5

CONCLUSION & FUTURE SCOPE

5.1 Conclusion

By combining picture tampering detection with image caption synthesis in an easily implemented manner, this integrated system marks a substantial improvement in image processing. Through the use of an Error Level Analysis (ELA) together with a Fine-tuned Convolutional Neural Network (CNN), we guarantee reliable image recognition and elimination of manipulated images, protecting the integrity of visual content. Additionally, our picture captioner, powered by Bidirectional Encoder Representations from Transformers (BERT) and vision models, improves accessibility by supplying evocative captions, thereby overcoming hurdles related to language and vision.

5.2 Future Scope

Since there remain significant obstacles to interfacing with the real world, including the public domain, the project has many paths for future research and development: refining detection algorithms to better highlight manipulations, perhaps incorporating deep learning architectures or coming up with more creative feature engineered; exploring more modalities to build an integrated multimedia analysis system; conducting large-scale deployment and assessment studies to assess the system's performance across multiple domains; carefully examining the ethical concerns around privacy, possible biases and other legal implications, and studying adversarial robustness and cross-domain applications to ensure a productive and robust integrated system for multiple real-world circumstances.

References

[1] S. McCloskey and M. Albright, “Detecting gan-generated imagery using saturation cues,” in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019.

[2] M. Qadir, S. Tehsin, and S. Kausar, “Detection of copy move forgery in medical images using deep learning,” in *2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)*. IEEE, 2021.

[3] J. Chen, J. Chen, W. Wang, and Y. Zhu, “Improved model for image tampering monitoring based on fast-rcnn,” in *2nd International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACAI)*, 2023.

[4] A. Mishra, K. T. Chui, and B. B. Gupta, “Tempered image detection using ela and convolutional neural networks,” in *IEEE International Conference on Consumer Electronics (ICCE)*. Hong Kong: IEEE, 2024.

[5] Y. William, S. Safwat, and M. A.-M. Salem, “Robust image forgery detection using point feature analysis,” in *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems*, 2019.

[6] J. Sudhakar, V. V. Iyer, and S. T. Sharmila, “Image caption generation using deep neural networks,” in *International Conference for Advancement in Technology (ICONAT)*, 2022.

[7] D. Sharma, D. Sharma, and D. Kumar, “Automated image caption generation framework using adaptive attention and bi-lstm,” in *IEEE Delhi Section Conference (DELCON)*. IEEE, 2022.

[8] S. R. Chandaran, S. Natesan, G. Muthusamy, P. K. Sivakumar, P. Mohanraj, and R. J. Gnanaprakasam, “Image captioning using deep learning techniques for partially impaired people,” in *International Conference on Computer Communication and Informatics (ICCCI)*. IEEE, 2023.

[9] I. Chhatbar, M. Gondhalekar, S. Pimple, and R. Pawar, “Machine interpretation of medical images using deep learning,” in *2nd Global Conference for Advancement in Technology (GCAT)*. IEEE, 2021.

[10] Y. Kim, J. Kim, and Y. M. Ro, “Mitigating dataset bias in image captioning through clip confounder-free captioning network,” 2023.

[11] S. Nakamura, H. Yanagimoto, and K. Hashimoto, “Movie caption generation with vision transformer and transformer-based language model,” in *2023 14th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*. Osaka, Japan: IIAI, 2023.

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. [Online]. Available: https://arxiv.org/abs/2010.11929

[13] F. Marra and C. D. V. L. Gragnaniello, "Detection of gan-generated fake images over social networks," 2018.

Proof of Publishing

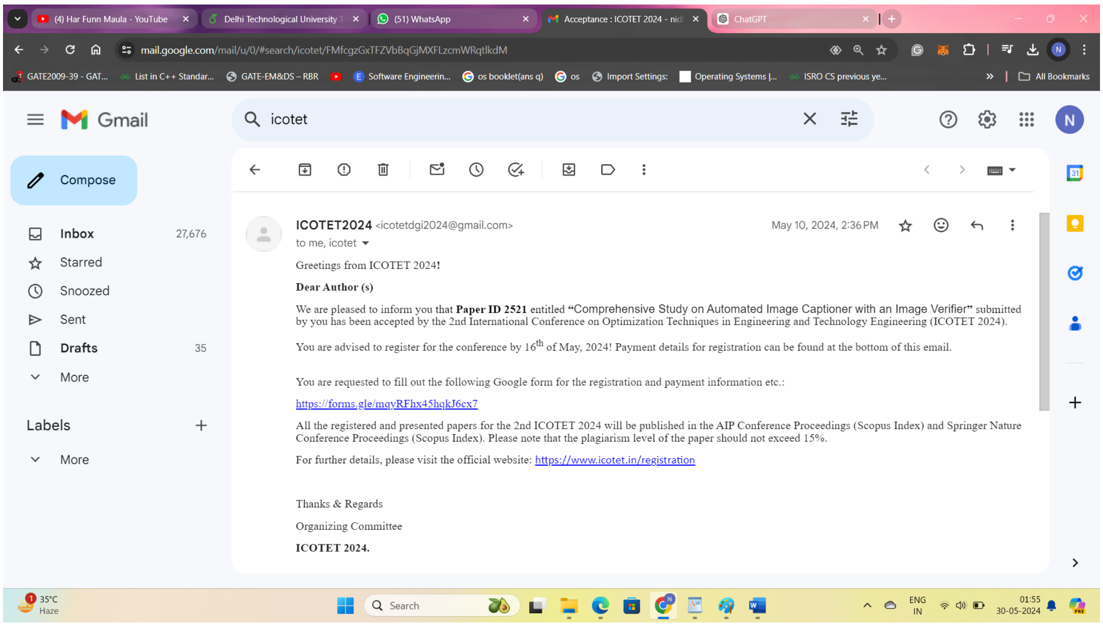
1st Paper

Title:

”Comprehensive Study on Automated Image Captioner with an Image Verifier”.

Status:

Accepted in ”2nd International Conference on Optimization Techniques in Engineering and Technology Engineering (ICOTET 2024)”.


(4) Har Funn Maula - YouTube
Delhi Technological University
(51) WhatsApp
Acceptance : ICOTET 2024 - nid
ChatGPT
mail.google.com/mail/u/0/#search/icotet/FMfcgzGxTFZVbBqGjMXFLzcmWRqtlkdM
Gmail
icotet
Compose
Inbox 27,676
Starred
Snoozed
Sent
Drafts 35
More
Labels
More
Search
35°C
Haze
ENG
IN
01:55
30-05-2024

ICOTET2024 <icotetdgi2024@gmail.com>

to me, icotet

May 10, 2024, 2:36 PM

Greetings from ICOTET 2024!

Dear Author (s)

We are pleased to inform you that **Paper ID 2521** entitled “Comprehensive Study on Automated Image Captioner with an Image Verifier” submitted by you has been accepted by the 2nd International Conference on Optimization Techniques in Engineering and Technology Engineering (ICOTET 2024).

You are advised to register for the conference by 16th of May, 2024! Payment details for registration can be found at the bottom of this email.

You are requested to fill out the following Google form for the registration and payment information etc.:

https://forms.gle/mqyRFhx45hqkJ6cx7

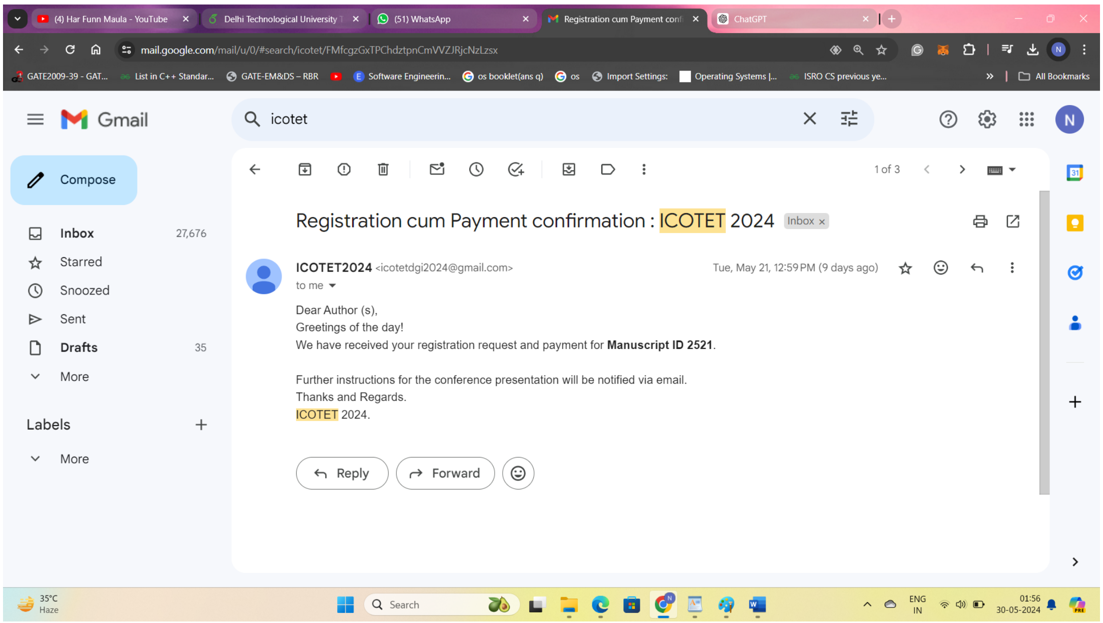
All the registered and presented papers for the 2nd ICOTET 2024 will be published in the AIP Conference Proceedings (Scopus Index) and Springer Nature Conference Proceedings (Scopus Index). Please note that the plagiarism level of the paper should not exceed 15%.

For further details, please visit the official website: https://www.icotet.in/registration

Thanks & Regards

Organizing Committee

ICOTET 2024.


(4) Har Funn Maula - YouTube
Delhi Technological University
(51) WhatsApp
Registration cum Payment conf
ChatGPT
mail.google.com/mail/u/0/#search/icotet/FMfcgzGxTPChdztpnCmVVZJRjcNzLzsx
Gmail
icotet
Compose
Inbox 27,676
Starred
Snoozed
Sent
Drafts 35
More
Labels
More
1 of 3

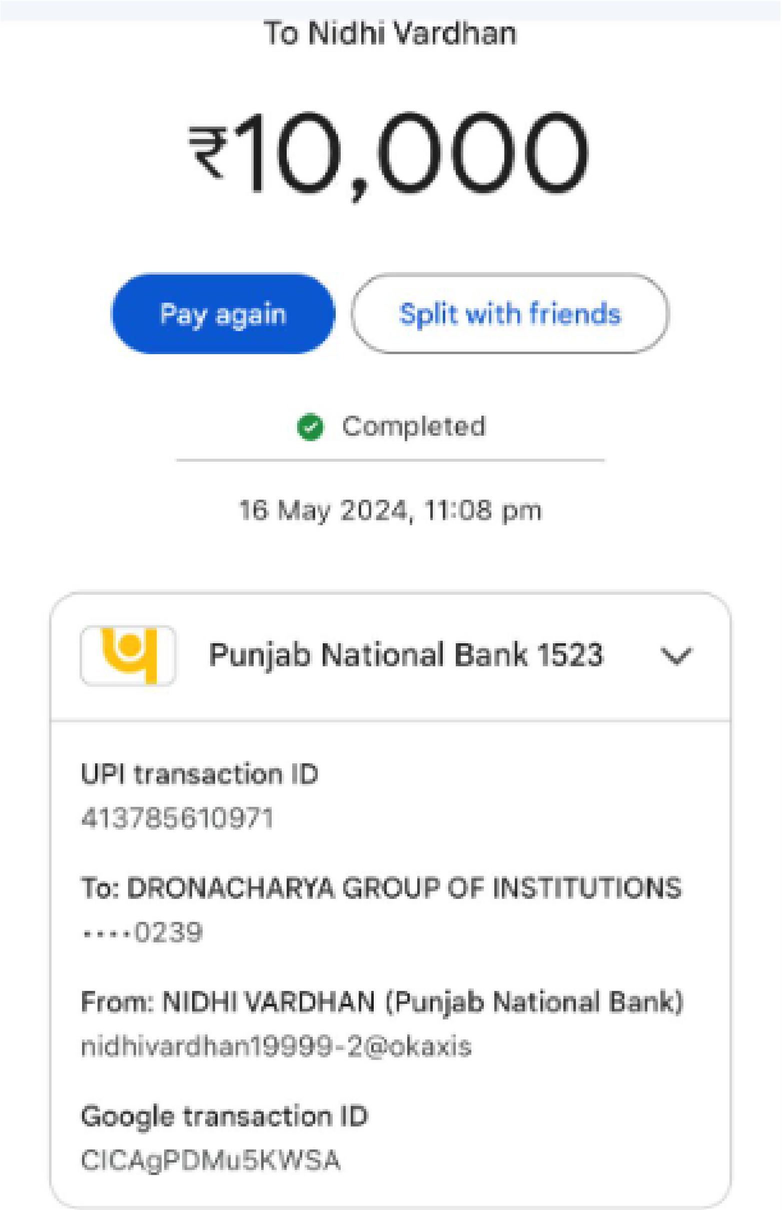
Registration cum Payment confirmation : ICOTET 2024 Inbox

ICOTET2024 <icotetdgi2024@gmail.com>
to me
Tue, May 21, 12:59 PM (9 days ago)

Dear Author (s),
Greetings of the day!
We have received your registration request and payment for **Manuscript ID 2521**.

Further instructions for the conference presentation will be notified via email.
Thanks and Regards.
ICOTET 2024.

Reply
Forward
35°C
Haze
Search
ENG
IN
01:56
30-05-2024


To Nidhi Vardhan

₹10,000

Completed

16 May 2024, 11:08 pm

Punjab National Bank 1523

UPI transaction ID
413785610971

To: DRONACHARYA GROUP OF INSTITUTIONS
••••0239

From: NIDHI VARDHAN (Punjab National Bank)
nidhivardhan19999-2@okaxis

Google transaction ID
CICAgPDMu5KWSA

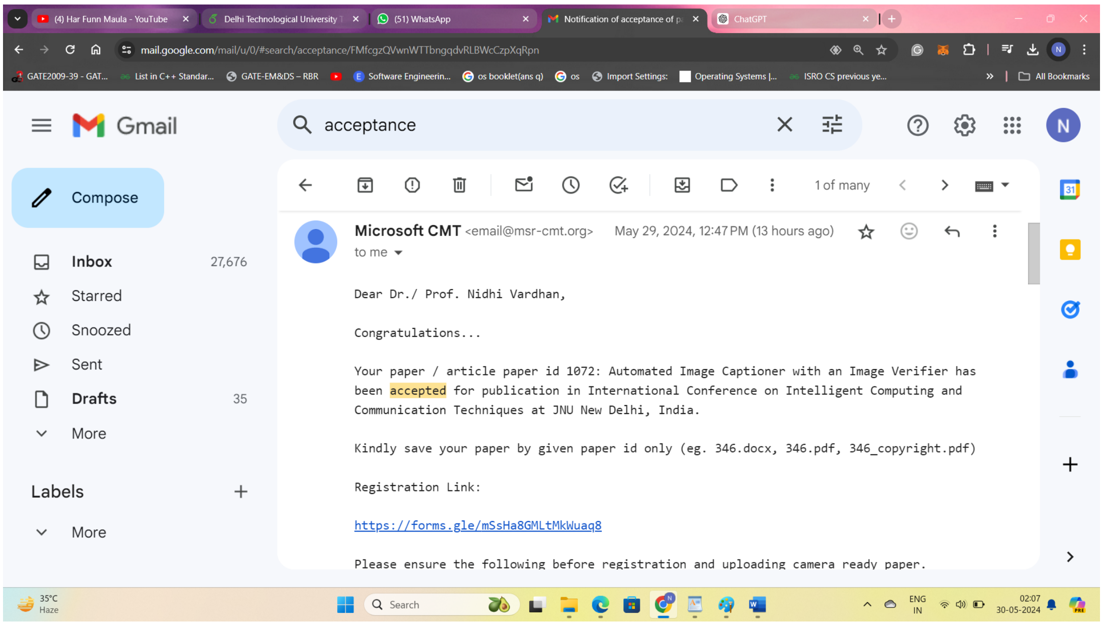
2nd Paper

Title :

”Automated Image Captioner with an Image Verifier”.

Status :

Accepted in ”International Conference on Intelligent Computing and Communication Techniques” at JNU New Delhi, India..



Microsoft CMT <email@msr-cmt.org> May 29, 2024, 12:47 PM (13 hours ago)
to me

Dear Dr./ Prof. Nidhi Vardhan,

Congratulations...

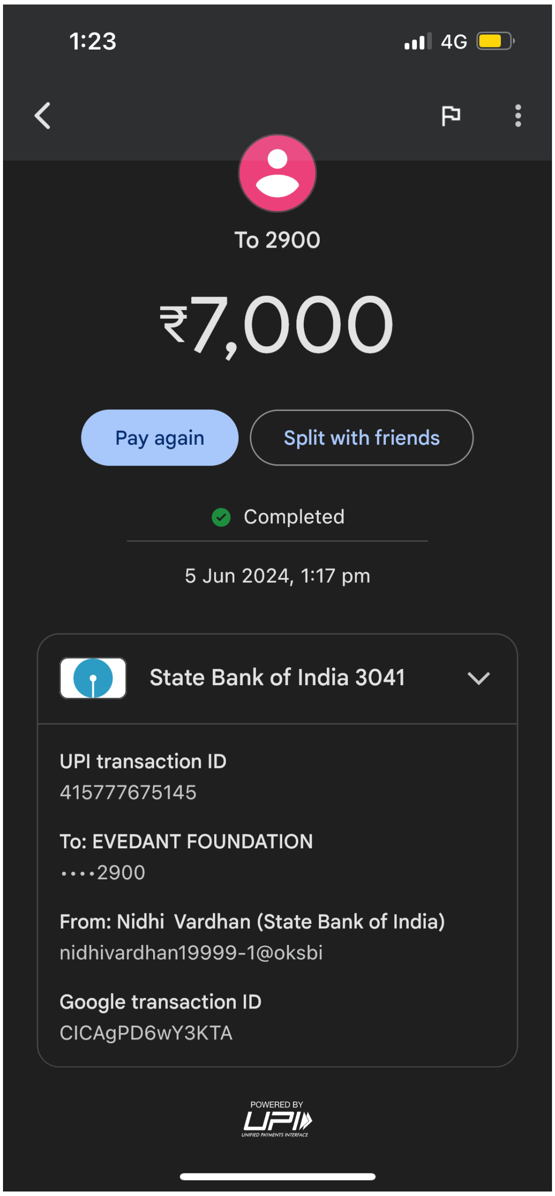
Your paper / article paper id 1072: Automated Image Captioner with an Image Verifier has been accepted for publication in International Conference on Intelligent Computing and Communication Techniques at JNU New Delhi, India.

Kindly save your paper by given paper id only (eg. 346.docx, 346.pdf, 346_copyright.pdf)

Registration Link:

https://forms.gle/mSsHa8GMLtMkWuaq8

Please ensure the following before registration and uploading camera ready paper.



To 2900

₹7,000

✓ Completed

5 Jun 2024, 1:17 pm

State Bank of India 3041

UPI transaction ID
415777675145

To: EVEDANT FOUNDATION
••••2900

From: Nidhi Vardhan (State Bank of India)
nidhivardhan19999-1@oksbi

Google transaction ID
CICAgPD6wY3KTA