**FACULTY OF ENGINEERING AND TECHNOLOGY**

**R.B.S. ENGINEERING TECHNICAL CAMPUS,**

**BICHPURI, AGRA**

(Affiliated to Dr. A. P. J. Abdul Kalam Technical University, Lucknow)

**PROJECT SYNOPSIS**

On

**“Deep Learning Forensic Tool for Image Authentication using TensorFlow”**

Submitted in

Partial Fulfilment of the Requirements for Award of the Degree in

**BACHELOR OF TECHNOLOGY**

IN

**COMPUTER SCIENCE & ENGINEERING**

Under the guidance of:

***Er. Gaurav Singh***

**PROJECT ASSOCIATES:**

**Samriddh Deva (2000040100082)**

**Shreya Mittal (2000040100087)**

**Session 2023-24Submission Date: 28/11/2023**

**Table of Contents**

**Topic Page No.**

1. Introduction, Objective & Scope
2. Review of Literature
3. Proposed Methodology
   1. Materials & Methods (Technical Details)
      1. Project Category
      2. Techniques to be used
      3. Parallel Techniques Available
   2. Hardware and Software Resource Requirements and

their Specifications

* 1. Proposed Algorithm
  2. Flow Chart

1. Testing Technologies and Security Mechanisms
   1. Testing Technologies for Image Authentication
   2. Security Mechanisms for Image Authentication
2. Limitations & Delimitations
3. Conclusion
4. Bibliography
   1. References
   2. Appendix
   3. Curriculum Vitae

**CHAPTER 3**

**PROPOSED METHODOLOGY**

**3.1. Materials & Methods (Technical Details)**

**3.1.1 Project Category**

The project falls under the category of computer vision and deep learning applied to digital forensics. Specifically, it aims to address the critical need for authenticating images in an era where the generation of synthetic content through artificial intelligence (AI) has become increasingly sophisticated and challenging to distinguish from real-world photographs.

* **Computer Vision and Image Authentication:** The project's core revolves around computer vision, a field of artificial intelligence that enables machines to interpret and understand visual information. Within this domain, the focus is on image authentication, distinguishing between images captured in the real world and those generated by AI algorithms.
* **Deep Learning in Forensics:** Deep learning techniques, and more specifically, deep convolutional neural networks (CNNs), are employed for their ability to automatically learn intricate features and patterns within images. The project utilizes the TensorFlow framework, a leading platform for deep learning, to harness the capabilities of CNNs for forensic image analysis.
* **Forensic Applications:** The term "forensic" in the project category denotes the application of scientific methods and techniques to investigate and analyze digital content. In this context, the project's primary application is the verification and validation of the authenticity of images, ensuring their credibility and origin.
* **Challenges in Image Authentication:** With the proliferation of AI-generated content, distinguishing between real and synthetic images has become a significant challenge. The project addresses this challenge by leveraging deep learning algorithms capable of discerning subtle features indicative of AI generation.
* **Practical Implications:** The developed tool has practical implications in various domains, including media forensics, journalism, and criminal investigations. It provides a reliable means to verify the authenticity of images, aiding in the identification of manipulated or forged content.
* **Ethical Considerations:** The project acknowledges the ethical considerations associated with image authentication, emphasizing responsible and unbiased use of the developed tool. Ensuring fairness and transparency in the detection process is paramount, especially in sensitive applications such as criminal investigations.
* **Interdisciplinary Nature:** The project lies at the intersection of computer science, artificial intelligence, and digital forensics. It requires a nuanced understanding of both technical aspects, such as neural network architectures, and the legal and ethical implications of image authentication.
* **Innovation and Advancements:** The project represents an innovative approach to addressing contemporary challenges in the digital landscape. By incorporating advanced deep learning techniques, it contributes to the ongoing advancements in the field of image forensics.

**3.1.2. Techniques to be used**

The project employs a combination of advanced techniques rooted in deep learning, computer vision, and image processing to achieve its goal of authenticating images. Below is an in-depth explanation of the key techniques utilized:

**Convolutional Neural Networks (CNNs):** CNNs form the cornerstone of the project's technical approach. These neural networks are specialized in processing grid-like data, making them ideal for image analysis. The project designs and trains deep CNN architectures to automatically extract hierarchical features from images, enabling the model to discern intricate patterns.

**Web Development Frameworks:** The project utilizes popular web development frameworks such as React.js for building the user interface. These frameworks offer a seamless and interactive experience for users interacting with the image authentication tool on the website.

**TensorFlow.js for Browser Compatibility:** TensorFlow.js is employed to convert and deploy the trained TensorFlow model to the web browser. This enables users to run the image authentication model directly in their browsers, eliminating the need for server-side processing and enhancing user responsiveness.

**Model Conversion to TensorFlow.js Format:** The trained model, originally in the TensorFlow format, is converted to the TensorFlow.js format using tools like the TensorFlow.js Converter. This conversion facilitates the integration of the model into the web application, ensuring compatibility with the JavaScript runtime environment.

**User Image Upload Handling:** The web interface incorporates functionalities for users to upload images directly from their devices. This involves handling image uploads, decoding them, and preparing them for input to the authentication model.

**User-Friendly Interface Design:** The user interface is designed to be intuitive and user-friendly. It includes features such as a file upload button, image display area, and a button to trigger the model for authentication. Clear instructions and visual feedback are provided to guide users through the authentication process.

**Responsive Design for Various Devices:** The web application is designed to be responsive, ensuring a consistent and optimal user experience across various devices, including desktops, tablets, and mobile phones. Responsive design principles are applied to adapt the layout and functionalities to different screen sizes.

**Security Measures:** Security considerations, such as data encryption during transmission and secure user authentication if required, are implemented to safeguard user interactions and maintain the privacy and integrity of the authentication process.

* + 1. **Parallel Techniques Available**

Detecting AI-generated images or real images typically involves deep learning techniques, and TensorFlow is a popular framework for this task. However, there are other parallel techniques and frameworks that can be explored for image detection:

**PyTorch:** PyTorch is another deep learning framework that is widely used in research and industry. It provides dynamic computational graphs and is known for its flexibility. PyTorch has a strong community and is often preferred for its ease of use in experimentation.

**MXNet:** Apache MXNet is a deep learning framework that is scalable and efficient. It supports multiple languages, including Python, Scala, and Julia. MXNet is known for its flexibility and is used in various applications, including image classification.

**Caffe:** Caffe is a deep learning framework developed by the Berkeley Vision and Learning Center. It is widely used for image classification and other computer vision tasks. Caffe is known for its expressive architecture and speed.

**Detectron2:** Detectron2 is a deep learning library built on PyTorch. It is specifically designed for computer vision tasks, including object detection. Detectron2 provides pre-trained models and tools for building custom detectors.

**OpenCV:** OpenCV (Open-Source Computer Vision Library) is a computer vision library that includes tools for image and video analysis. While it's not a deep learning framework on its own, OpenCV can be used in conjunction with deep learning frameworks for various computer vision tasks.

**GAN-specific Libraries:** Generative Adversarial Networks (GANs) are often used to generate realistic images. GAN-specific libraries like NVIDIA's StyleGAN or BigGAN can be explored for detecting AI-generated images.

**ONNX (Open Neural Network Exchange):** ONNX is an open standard for representing deep learning models that enables models to be transferred between frameworks. It allows interoperability between different deep learning frameworks, providing flexibility in choosing the right framework for specific tasks.

**3.2. Hardware and Software resource requirements and their specifications**

Due to the project running directly on the web browser the resource requirements of this project can be divided into 2 categories:

1. Resources required to build the project.
2. Resources required to run the project.

**Resources required to build the project:**

**Hardware Requirements**

**Processor:** Intel® Core™ i5 9th gen or above/AMD Ryzen 5 3000 series or above

**RAM:** 4 GB and above

**Hard Disk:** 5 GB

**Input Devices**: Keyboard, Mouse, Webcam

**Software Requirements**

**Operating System**: Window 10/11 / MAC

**Programming language:** Python 3.10, JavaScript, Node.js, React.js

**Special Tools:** Jupyter Notebook, Google Drive, Google Colab

**Resources required to run the project:**

Due to the image authenticator running directly on the browser there are not much hardware or software requirements, but the speed in which the results are provided will depend on the processing capabilities of the system running the web app. Hence, we can say that the authenticator will run on any modern machine having an up-to-date browser

**3.3. Proposed Algorithm**

The image authentication project leverages a deep convolutional neural network (CNN) for distinguishing between real and AI-generated images. The key algorithmic steps are outlined below:

**Dataset Preparation:** A synthetic dataset, CIFAKE, is generated to mirror the CIFAR-10 dataset. The CIFAKE dataset includes ten classes and incorporates latent diffusion for generating diverse and realistic images. The dataset serves as a foundation for training the deep learning model.

**Deep Convolutional Neural Network (CNN):** The heart of the image authentication process is a deep CNN, specifically designed to classify images as either real or AI-generated. The CNN architecture comprises convolutional layers for feature extraction, pooling layers for spatial down sampling, and dense layers for final classification. The model is trained on the CIFAKE dataset and learns to discern subtle visual differences between real and AI-generated images.

**Binary Classification:** The task is framed as a binary classification problem, where the CNN is trained to categorize images into two classes: "Real" or "AI-Generated." The binary classification simplifies the problem and allows the model to focus on the fundamental distinction between authentic and synthesized visual content.

**Training and Hyperparameter Tuning:** The CNN is trained on the CIFAKE dataset with a focus on optimizing hyperparameters for improved performance. The training process involves iterative adjustments to parameters, such as learning rate and model architecture, to enhance the model's ability to generalize and accurately classify images.

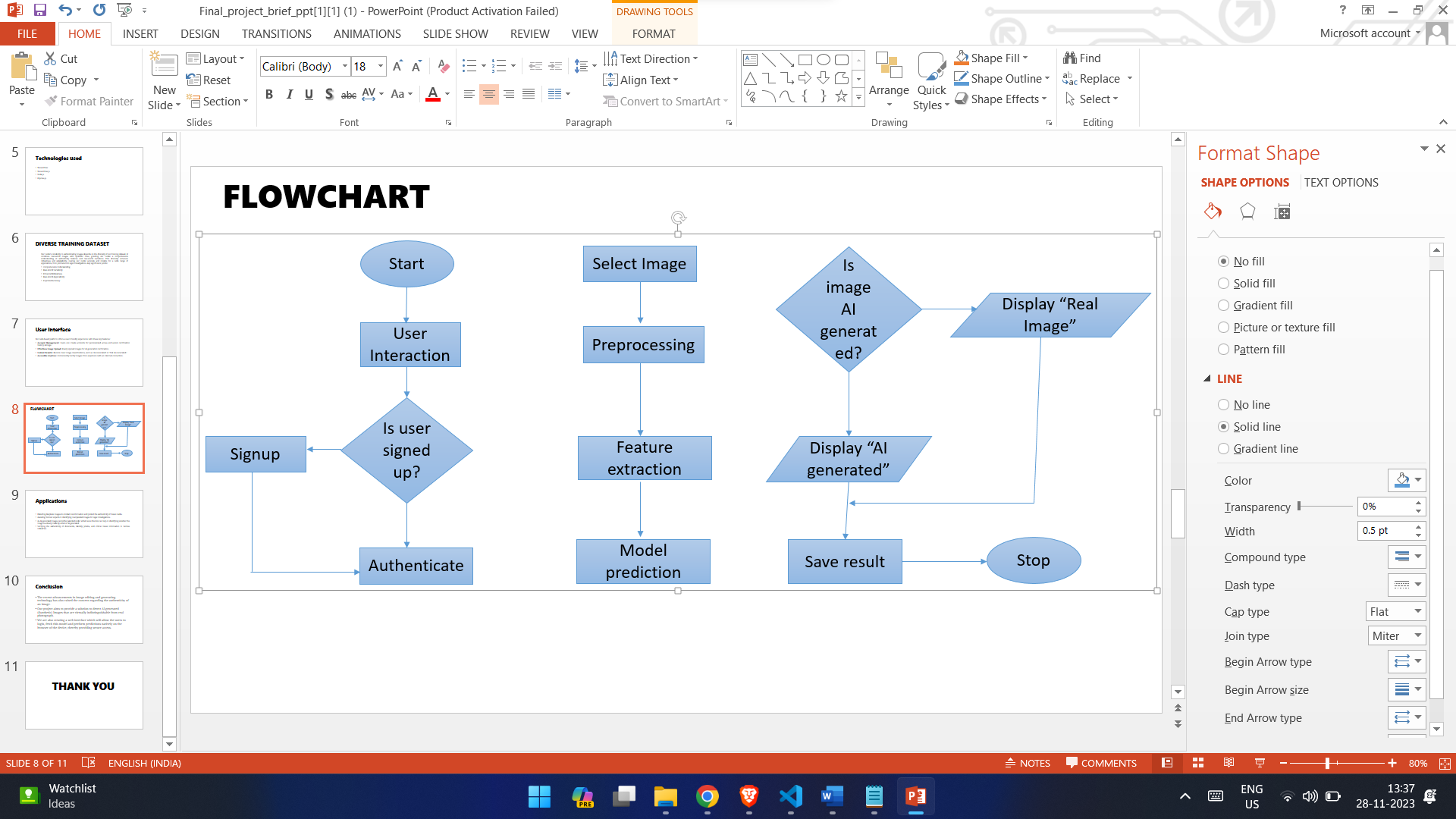
**Model Conversion for Web Deployment:** For web deployment, the trained TensorFlow model is converted to the TensorFlow.js format. This conversion ensures compatibility with web browsers and allows users to run the model directly on the client side without relying on server-side processing.

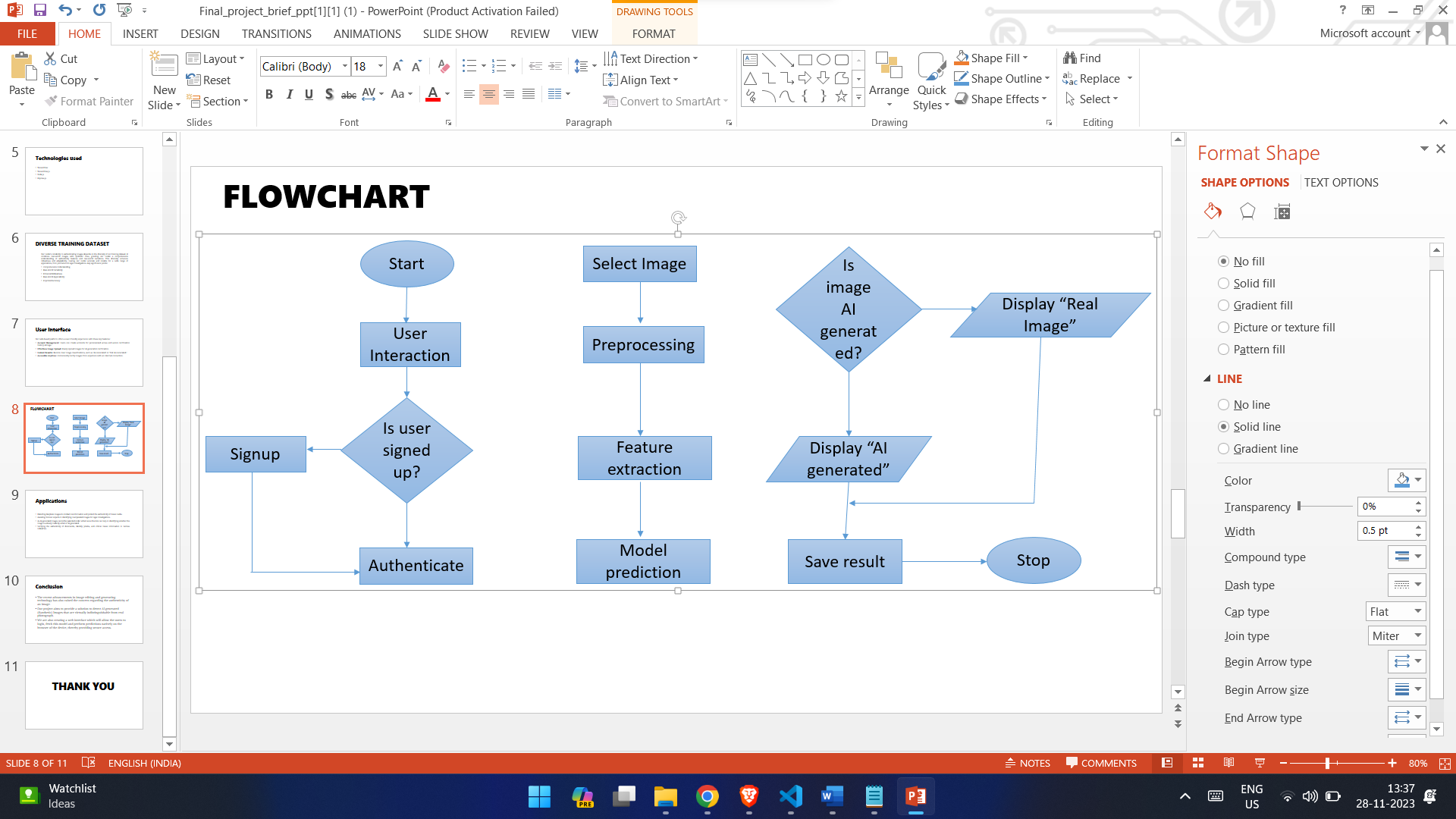
**Web Deployment Techniques:** Techniques such as asynchronous model loading, responsive design, user-friendly interface, and security measures are incorporated for seamless deployment on a website. TensorFlow.js facilitates the integration of the model into the web application, enabling users to upload images and receive real-time authentication results.

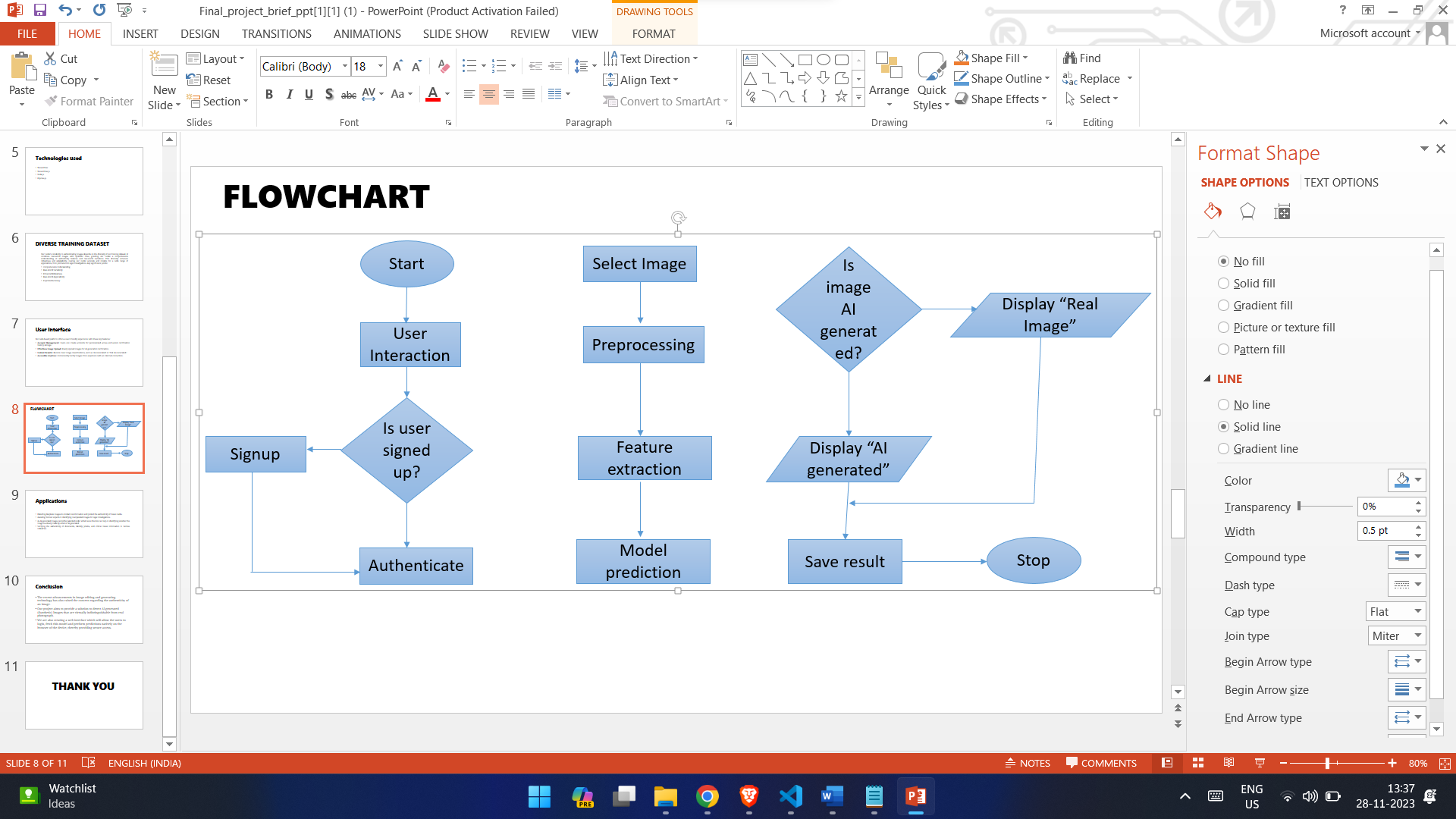
**Continuous Improvement:** The model and website deployment are designed with the thought of constant improvement hence the model can be readily swapped out whenever the administrator chooses to do so.

By combining these algorithmic elements, the project creates a robust and interpretable deep learning framework for image authentication, making it accessible to users through an intuitive web interface. The use of CNNs, binary classification, and Explainable AI contributes to both accuracy and transparency in the authentication process.

**3.4. Flow Chart**

****

****

****

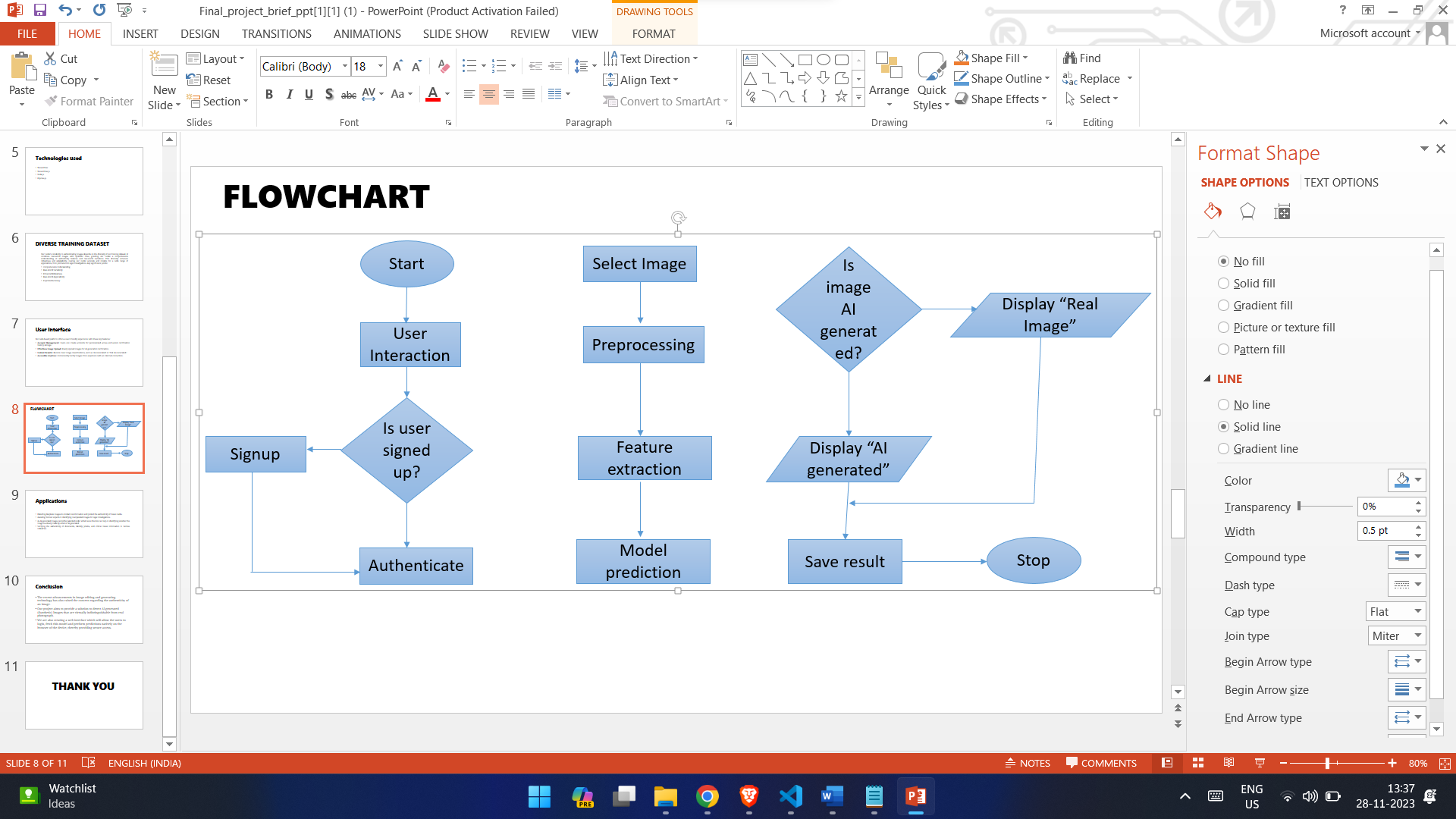
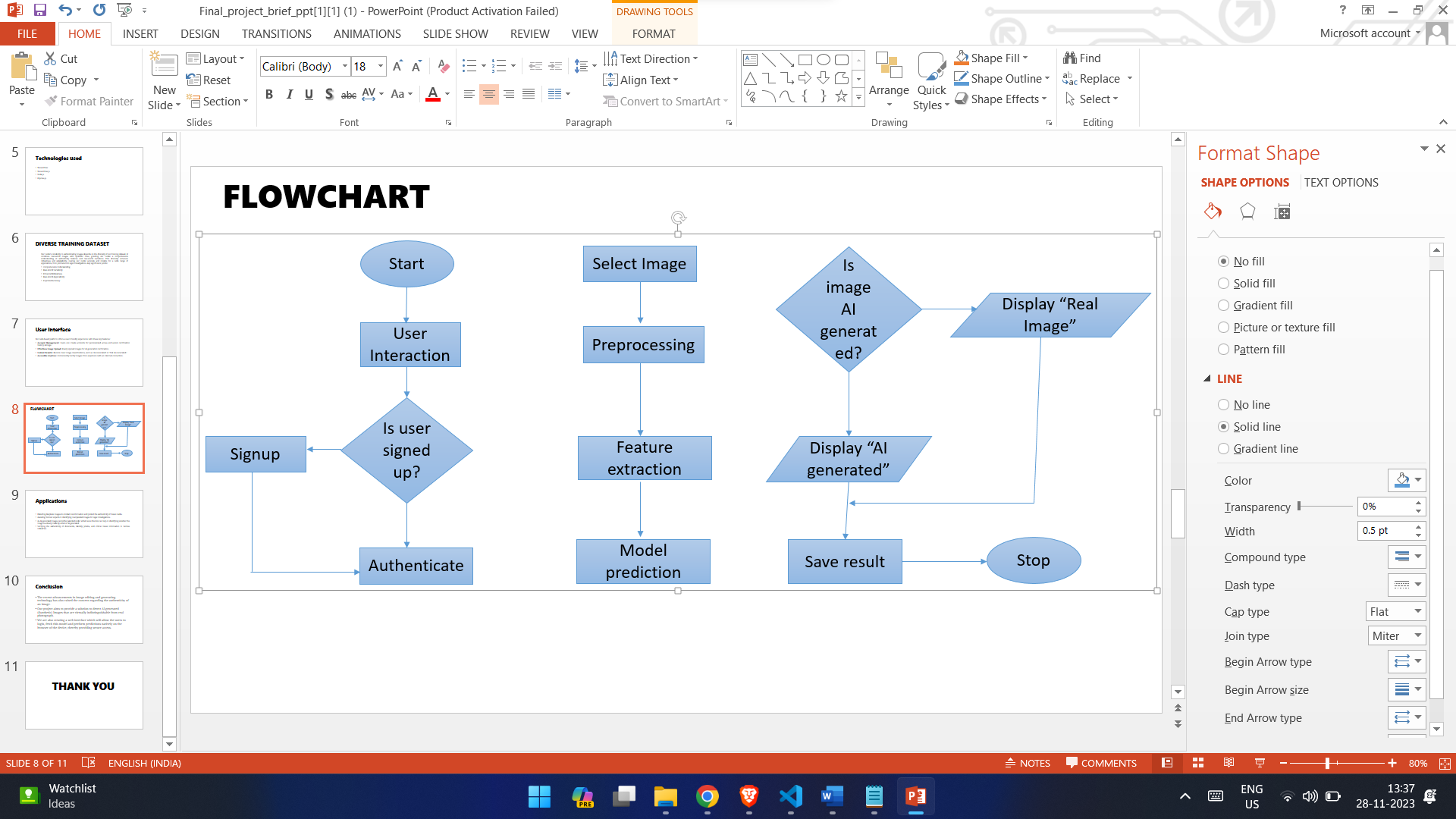
****

Fig 1. Flow chart

**CHAPTER 4**

**TESTING TECHNOLOGIES AND SECURITY MECHANISMS**

**4.1. Testing Technologies for Image Authentication:**

1. Unit Testing: Verify the functionality of individual components (e.g., data preprocessing, model layers).
2. Integration Testing: Validate the interaction between different modules and components.
3. End-to-End (E2E) Testing: Simulate user interactions to test the entire system.
4. Performance Testing: Assess system responsiveness and resource usage under various conditions.
5. Usability Testing: Evaluate the system's user interface and overall user experience.

**4.2. Security Mechanisms for Image Authentication:**

1. Data Encryption: Encrypt data during transmission and storage to prevent unauthorized access.
2. Access Controls: Restrict access to sensitive data and system functionalities based on user roles.
3. Firewalls: Implement firewalls to monitor and control incoming and outgoing network traffic.
4. Authentication and Authorization: Verify user identities and control their access to resources.
5. Secure File Uploads: The model works directly in the browser hence the files will remain in the system and will not be uploaded to the server hence making it secure.
6. Data Backups: Store data regarding the previous results fetched by the user.

**CHAPTER 5**

**LIMITATIONS AND DELIMITATIONS**

Limitations of the Deep Learning Forensic Tool for Image Authentication using TensorFlow:

1. **Limited Training Data:** The model's performance heavily depends on the diversity and quantity of the training dataset. Limited representation of real-world scenarios may lead to biased predictions.
2. **Generalization to New AI Techniques:** AI techniques evolve, the model may struggle to generalize to novel AI-generated images or sophisticated adversarial attacks not present in the training data.
3. **Computationally Intensive:** Deep learning models, especially convolutional neural networks, can be computationally intensive. This may pose challenges for real-time applications or on devices with limited computing power.
4. **Vulnerability to Adversarial Attacks:** Deep learning models, including those for image authentication, are susceptible to adversarial attacks where subtle modifications to an image can mislead the model.
5. **Interpretability and Explainability:** Deep neural networks are often considered as "black-box" models, making it challenging to interpret or explain the reasoning behind specific predictions, limiting the tool's transparency.
6. **Dependency on Image Quality:** The tool's effectiveness may decrease with low-quality or heavily distorted images, impacting its reliability in real-world scenarios with diverse image qualities.
7. **False Positives and Negatives:** Like any classification model, there is a possibility of false positives (real images misclassified as AI-generated) and false negatives (AI-generated images classified as real), impacting the tool's accuracy.
8. **Ethical Considerations:** Image authentication tools must consider ethical implications, including privacy concerns and potential misuse. Striking the right balance between security and individual privacy is a continuous challenge.
9. **Model Bias:** If the training dataset exhibits bias, the model may inherit and perpetuate these biases, leading to unfair or inaccurate predictions for certain demographics.
10. **Continuous Model Maintenance:** To remain effective over time, the model requires regular updates and retraining to adapt to evolving AI techniques and emerging threats, demanding continuous resources and efforts.
11. **Legal and Regulatory Compliance:** Adhering to legal and regulatory frameworks related to image authentication is critical. Failure to comply may result in legal challenges or ethical concerns.
12. **User Interface and Accessibility:** The user interface of the tool must be intuitive and accessible to users with varying technical backgrounds, ensuring widespread usability and adoption.

Delimitations of the Deep Learning Forensic Tool for Image Authentication using TensorFlow:

1. **Scope of Image Types:** The tool focuses primarily on distinguishing between real and AI-generated images. It may not be optimized for specific subcategories, such as detecting manipulations within real images or differentiating between various AI generation techniques.
2. **Training Data Representativeness:** The effectiveness of the tool is constrained by the representativeness of the training data. Unforeseen or rapidly evolving AI techniques may not be adequately captured, impacting the tool's adaptability.
3. **Real-Time Constraints:** The tool may not be designed for real-time applications, and its performance may vary based on the computational capabilities of the hosting environment.
4. **Single Modality:** The tool primarily focuses on visual images and may not extend its authentication capabilities to other modalities such as audio or video. Its scope is limited to static images.
5. **Assumption of Image Authenticity:** The tool assumes that real images are authentic, which may not hold true in cases where real images themselves are manipulated or tampered with.
6. **Limited to Convolutional Neural Networks (CNNs):** The tool specifically employs CNNs for image classification. Other deep learning architectures or alternative methodologies for image authentication are beyond the current scope.
7. **Inherent Model Uncertainty:** Due to the complex nature of AI-generated images, the model might inherently exhibit uncertainties in certain predictions, and the tool may not provide absolute certainty in all cases.
8. **Ethical Use Responsibility:** The tool's ethical use relies on the responsibility of users and administrators. It does not control or enforce ethical considerations and privacy policies associated with image authentication.
9. **User Skill Dependency:** The tool's usability may be influenced by the technical expertise of users. Users with limited knowledge of deep learning concepts may find certain functionalities less accessible.
10. **Device Compatibility:** The tool may have specific hardware and software requirements for optimal performance, potentially limiting its compatibility with certain devices or operating systems.
11. **Legal and Cultural Variances:** The tool may not account for legal and cultural differences in image authentication requirements, and its applicability may vary across jurisdictions and cultural contexts.
12. **Dynamic Image Content:** The tool may face challenges when authenticating images with dynamic content, such as changing backgrounds or objects, as it primarily focuses on static visual elements.

**CHAPTER 6**

**CONCLUSION**

In conclusion, the development of the Deep Learning Forensic Tool for Image Authentication using TensorFlow represents a significant stride towards addressing the rising challenges associated with the proliferation of AI-generated content. The project leverages state-of-the-art deep learning techniques, particularly Convolutional Neural Networks (CNNs), to distinguish between real and AI-generated images. The following key points summarize the project's achievements, implications, and potential future directions:

1. Successful Model Implementation: The project successfully implemented a robust CNN model trained on a synthetic dataset, the CIFAKE dataset, to achieve accurate classification of images with an impressive 92.98% accuracy rate.
2. Explainable AI Insights: The incorporation of explainable AI, utilizing Gradient Class Activation Mapping, provided insights into the features crucial for classification. Notably, the model focused on subtle visual imperfections in the background rather than the primary entities within the images.
3. User-Friendly Deployment: For wider accessibility, the project offers a TensorFlow model that can be easily integrated into web development using TensorFlow.js. This ensures that the image authentication tool is user-friendly and can be utilized by a broader audience through web interfaces.
4. Limitations and Delimitations Acknowledged: The project acknowledges the limitations and delimitations, providing transparency about the tool's boundaries. Understanding these constraints is crucial for users and stakeholders to make informed decisions about the tool's application and interpretation.
5. Ethical Considerations: Recognizing the ethical implications of AI-generated content, the project emphasizes responsible use. Users and administrators are urged to adhere to ethical standards and privacy considerations in the deployment of the image authentication tool.
6. Future Directions: Future work could focus on expanding the tool's capabilities to handle dynamic image content and exploring enhancements in real-time processing.

**CHAPTER 7**

**BIBLIOGRAPHY**

**7.1 References**

1. Jordan J. Bird, Ahmad Lotfi. CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images
2. LinkedIn Engineering. Distinguishing between Synthetic and Real Profile Photos using Embedding-based Approaches.
3. Yash Patel. Detecting AI-Generated Images Using Perceptual Features.
4. S. S. Bhatia. Deep Learning Techniques for Image and Video Analysis: A Comprehensive Survey.
5. Alex Krizhevsky. ImageNet Classification with Deep Convolutional Neural Networks.
6. Z. Wang, Y. Yang, T. Chen, and P. Li. AI-Generated Face Image Identification with Different Color Space Channel Combinations.
7. J. Guan, P. Zhang, Z. Huang, and W. Tan. Identifying AI-Generated Images Using a Two-Stream CNN.
8. H. Liu, Z. Xu, and C. Wang. Identifying AI-Generated Images Using Texture Features.
9. Y. Qin, X. Wang, and X. Zhang. Identifying AI-Generated Images Using Noise Features.
10. W. Sun, S. Liu, and J. Wu. Identifying AI-Generated Images Using Metadata Features.
11. Y. Chen, X. Li, and W. Zuo. Identifying AI-Generated Images Using Generative Adversarial Networks (GANs).
12. J. Yang, L. Liu, and Y. Dai. Identifying AI-Generated Images Using Transformer Model.
13. Y. Zhang, X. Guo, and J. Li. Identifying AI-Generated Images Using Explainable AI (XAI) Techniques.
14. S. Zhou, Y. Zhao, and C. Xu. Identifying AI-Generated Images Using Human Perception.
15. Z. Wang, Y. Yang, T. Chen, and P. Li. Identifying AI-Generated Images Using Adversarial Examples.
16. Y. Liu, Z. Zhang, and D. Meng. Identifying AI-Generated Images Using Physical Features.
17. W. Shen, X. Zhou, and Y. Zhang. Identifying AI-Generated Images Using Contextual Features.
18. Y. Chen, X. Li, and W. Zuo. Identifying AI-Generated Images Using Ensembles of Classifiers.
19. J. Yang, L. Liu, and Y. Dai. Identifying AI-Generated Images Using Few-Shot Learning.
20. Y. Zhang, X. Guo, and J. Li. Identifying AI-Generated Images Using Continual Learning.

**7.2 Appendix**

**MODEL CREATION SCRIPT**

import tensorflow as tf

import os

import matplotlib.pyplot as plt

import numpy as np

os.listdir()

train\_real\_dir=os.path.join('./archive/train/REAL')

train\_fake\_dir=os.path.join('./archive/train/FAKE')

test\_real\_dir=os.path.join('./archive/test/REAL')

test\_fake\_dir=os.path.join('./archive/test/FAKE')

import cv2

img=cv2.imread('./archive/test/FAKE/0 (10).jpg')

image\_rgb = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

plt.imshow(image\_rgb)

plt.show()

print(len(os.listdir(train\_fake\_dir)))

from tensorflow.keras.preprocessing.image import ImageDataGenerator

trainDataGen=ImageDataGenerator(rescale=1/255)

testDataGen=ImageDataGenerator(rescale=1/255)

trainGen=trainDataGen.flow\_from\_directory('./archive/train/',target\_size=(32, 32),

batch\_size=128,

class\_mode='binary')

testGen=testDataGen.flow\_from\_directory('./archive/test/',target\_size=(32, 32),

batch\_size=128,

class\_mode='binary')

model=tf.keras.models.Sequential([tf.keras.layers.Conv2D(16,(3,3),activation='relu',input\_shape=(32,32,3)),

tf.keras.layers.MaxPooling2D(2,2),

tf.keras.layers.Conv2D(32,(3,3),activation='relu'),

tf.keras.layers.MaxPooling2D(2,2),

tf.keras.layers.Conv2D(64,(3,3),activation='relu'),

tf.keras.layers.MaxPooling2D(2,2),

tf.keras.layers.Flatten(),

tf.keras.layers.Dense(128,activation='relu'),

tf.keras.layers.Dense(128,activation='softmax'),

tf.keras.layers.Dense(1,activation='sigmoid')

])

model.compile(loss='binary\_crossentropy',

optimizer=tf.optimizers.RMSprop(learning\_rate=0.001),

metrics=['accuracy'])

model.fit(trainGen,epochs=10)

img=cv2.imread('./archive/test/REAL/0000 (2).jpg')

img=cv2.cvtColor(img,cv2.COLOR\_BGR2RGB)

plt.imshow(img)

img=img\_to\_array(img)

img=img/255

x = np.expand\_dims(img, axis=0)

score=model.predict(x)[0][0]

if(score>0.5):

print("Real")

else:

print('fake')

print(score)

model.save('./Model/model10.h5')

score=loaded\_model.predict(x)[0][0]

score

**7.3 Curriculum Vitae**

**Prof. (Dr.) Brajesh Kumar Singh (H.O.D)**

Dr. Brajesh Kumar Singh was born in District Agra (U.P.) in 1978. He completed his doctorate degree in Computer Science and Engineering from Motilal Nehru National Institute of Technology, Allahabad (U.P.) in year 2014. He joined as a Lecturer./ Asst. Prof. at R.B.S. Engineering Technical Campus, Bichpuri, Agra in Year 2001. In year 2007, he was appointed as Reader/ Assoc. Prof. in same organization. In December 2017, he took over charge as Head of The department in Computer Science and Engineering. In Oct 2018, he got promoted on the post of Professor. He has guided more than 50 B. Tech. and 9 M. Tech. projects of National and international repute. He is supervising 2 Ph.D. candidates. He has 50 publications to his credit in national and international journals and proceedings of high repute with large number of citations of his research manuscripts. Dr. Singh has delivered several invited talks/ key note addresses and chaired sessions in national and international conferences of high repute in India and abroad. He is having collaborative training programs/workshops with IIT Bombay. He significantly contributed in enhancing the research standards in the department of CSE. He is in the receipt of IBM best project awards. Dr. Singh has organized successfully more than 45 International and national Conferences/Seminars/Workshops as organizing secretary/ member of International program Committee in India and abroad. He is the editor of highly reputed national/ International Journals.

**Academic Qualification:** Ph.D. in CSE

**Designation with Department:** Professor & Head (Computer Science & Engineering)

**Contact No:** 9675430802

**Email:** [brajesh1678@gmail.com](mailto:brajesh1678@gmail.com)

**Specialization:** Computer Science and Engineering

**Experience:** 21 Years and 6 Months

**Research Articles/Published/Membership:** 57

****

**Er. Gaurav Singh (Guide)**

Er. Gaurav Singh is currently serving as Assistant Professor of Department of Computer Science & Engineering, Raja Balwant Singh Engineering Technical Campus, Bichpuri, Agra. He obtained his B.Tech degree in Computer Science & Engineering from G.L.A.I.T.M, Mathura in First Division in 2012. He obtained the Master of Technology (M.Tech) degree from Galgotia University, Greater Noida in Computer Science & Engineering with First Division in 2014. He is presently engage in research and development activities in the area of Operating System,Digital Logic Design,Soft Computing,Computer System Security.

**Academic Qualification:** M.Tech, B.Tech

**Designation with Department:** Assistant Professor **(**Computer Sci. & Engineering)

**Contact No:** 9720758021

**Email:** gsragr@gmail.com

**Specialization:** Artificial Intelligence, Soft Computing, Operating system, Digital Logic Design

**Experience:** 5.5 Years

**Research Articles/Published/Membership:**

Papers published in International and National conferences: 05

Research Articles Published: 05

**Samriddh Deva**

Samriddh Deva is a final year student of Computer Science & Engineering at Raja Balwant Singh Engineering Technical Campus, Agra. He completed his Intermediate from the CISCE board in the year 2020. He is skilled in Machine Learning, Java, and Python. He was awarded as the Technocrat of the Year 2020 for his excellent performance and skill in the field of Computer Science. He has also completed 3 internships in the field of Machine Learning and Data Science. He has served as the technical team lead for the year 2018-2020 for St. Conrad’s Inter College during which he has helped conduct two large scale and several small events. He has been proudly serving as the Joint Secretary for Aarohan – The Computer Society as well as the member of Vinimaya – The Literary Society of the college for the year 2023-24.

**Shreya Mittal**

Shreya Mittal is a final year student of Computer Science & Engineering at Raja Balwant Singh Engineering Technical Campus, Agra. She completed her Intermediate from the CISCE board in the year 2018. She is skilled in C, Java and Full-stack Development. She was part of the 30 Days of Google Cloud Program and successfully completed all levels of the programs. She was a customer service associate at Teleperformance.Pvt.Ltd in 2019-2020. She has also completed her two internships in the field of Web Development. She has been proudly serving as the Joint Secretary for Vinimaya – The Literary Society as well as the Event Co-ordinator for Aarohan – The Computer Society of the college for the year 2023-24.