### 1. Project Title: Applying Computer vision for Identification of Real vs. AI-Generated Fake Images

This project proposal is prepared for UMBC Data Science Master Degree Capstone by Dr Chaojie (Jay) Wang.

* Author Name: Avanti Dasre
* Link to the author’s GitHub profile: https://github.com/ad-04
* Link to the author’s LinkedIn profile: https://www.linkedin.com/in/avanti-dasre-7a3a8b152/
* Link to your PowerPoint presentation file : https://1drv.ms/p/s!AlbITO6Fex1YgQhu6Jb6UV7Fg7qx
* Link to your YouTube video

### 2. Introduction

In today’s digital age, the proliferation of manipulated or fake images has become a pressing concern, influencing public perception, misinforming individuals, and eroding trust in visual media. This project addresses the urgent need for effective fake image detection, leveraging advancements in computer vision, artificial intelligence, and deep learning. As images increasingly shape our understanding of the world, the ability to distinguish authentic content from manipulated ones is paramount. The project aims to develop and evaluate sophisticated algorithms capable of identifying forged images, be it through digital alterations, deep fakes, or other techniques. The implications of this research extend to various fields, from journalism and forensics to social media and cybersecurity, where discerning truth from deception is vital. By exploring the methods, challenges, and societal impact of fake image detection, this report offers a comprehensive understanding of this critical area of study and underscores the importance of technological solutions in the battle against visual misinformation.

## Problem Statement & Background:

The primary objective of this project is to develop a deep learning model that can accurately distinguish between real and AI-generated images. With the rapid advancements of generative artificial intelligence models and the convincing nature of fake images, it is becoming increasingly difficult for humans to distinguish them from the real ones. This is a challenging problem as the increasing quality of AI-generated images has led to concerns regarding the authenticity and trustworthiness of visual content. This project aims to address this problem with effective use of computer vision and deep neural networks to detect whether an image is real or has been generated by AI?

By end of this project, I aim to explore & find answers to following research questions: - Can a machine learning model accurately distinguish between real and AI-generated images based on underlying visual features? - What are the key visual features or cues that differentiate real images from AI-generated ones? - How does the performance of computer vision algorithms based on convolutional neural networks (CNNs) vary in detecting AI-generated mages? - How do various image processing techniques and data augmentation techniques influence the model performance? - How do different pretrained models help to improve the accuracy of the model?

Potential real-world use cases of this project include - Content Moderation by social media websites and internet companies. - Media organizations can use this model to verify authenticity of images and videos to combat the spread of fake news. - This model can be expanded to similar domain specific use cases such as Digital forensics & counterfeit product detection.

### 3. Dataset

Data Sources: The CIFAKE dataset[1] is created and published in the year 2023. The dataset contains 60,000 synthetically-generated images and 60,000 real images (collected from CIFAR-10).

Dataset link: https://www.kaggle.com/datasets/birdy654/cifake-real-and-ai-generated-synthetic-images Dataset Stats: - The dataset contains two classes - REAL and FAKE. - Number of train image samples - 1,00,000 - Number of test image samples - 20,000 - Data size: Around 120 MB

This dataset appears to be designed for tasks related to distinguishing between real and synthetically generated images, which can be valuable in the context of image classification, deep learning, or computer vision research. Researchers and practitioners can use this dataset for various purposes, such as developing and testing algorithms for image classification and synthetic image detection.

## Steps:

1.Data collection: The images are loaded into the runtime directly from kaggle using the kaggle library. The images are segregated into folders labeled with respective labels The images are loaded into numpy arrays using the opencv library While EDA we can find that there are 50 thousand real and 50 thousand fake images.

2.Data preparation: The images are loaded into numpy arrays and resized to a standard size of 32x32x3 using opencv

3.Model building: Tensorflow and keras libraries are used to build the models Three models have been built

### 4. Exploratory Data Analysis

* Analyze/Plot distribution of the target classes in train and test datasets.
* Use python based visualization frameworks like numpy, opencv to load & preprocess the images.
* Use matplotlib to load and view the images samples for both real & fake classes.
* Augment the input data using techniques such as random resizing, cropping, or rotating.

### 5. Data preparation & Model Building

1. Preprocessing and data splitting: Divide the dataset into training and validation sets using random splitting. Ascertain random splitting unless there is an imbalance in the classes. Load data by loading pictures as arrays. Standardize the picture arrays by data standardization and resizing. Adjust the image sizes to a standard size.
2. Image Augmentation: Use a variety of ways to enhance images, such as rotating, resizing, and random cropping. By exposing the model to a wider range of data, this improves its ability to generalize.
3. CNN Architectures: Create two or three distinct CNN architectural designs. Try out several convolutional and pooling layers to determine which model performs the best.
4. Pre-trained Models: Try out pre-trained models such as Densenet, ResNet, and VGG. Adjust the classification layer to make the model more suitable for the given goal.
5. Model Training and Visualization: Use this method to train the models.Utilizing the training dataset, train the models, then validate them using the validation dataset. To see the model’s loss on training and validation data, make charts.
6. Evaluation: To precisely gauge the finished model’s performance, assess it using a test dataset.
7. Confusion Matrix: To determine where the model is erroneous, use a confusion matrix. This aids in pinpointing the particular classes that the model finds challenging.
8. Gradio-Enabled Web Application: Use pip install gradio to install Gradio. Using Gradio, create a basic online application that allows users to submit images. To create predictions, include the learned model into the online application. For example photographs, show both the expected and actual classes.
9. Deployment: Set up a server to host the web application. Gradio has integrated support for launching apps across many platforms.
10. User Feedback and Interaction: Make it possible for users to communicate with the model via the web interface. Gather user input to enhance the program or the model.
11. Maintenance and Monitoring: Establish monitoring to keep tabs on the functioning of the deployed model. To keep the model accurate, add fresh data to it on a regular basis.
12. Documentation: Clearly explain to users how to utilize the web application in your clear documentation.

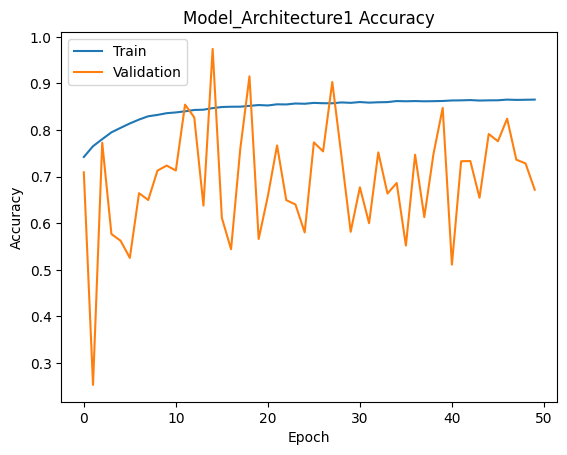
## Modeling Building

Tensorflow and keras libraries are used to build the models. Three models have been built :

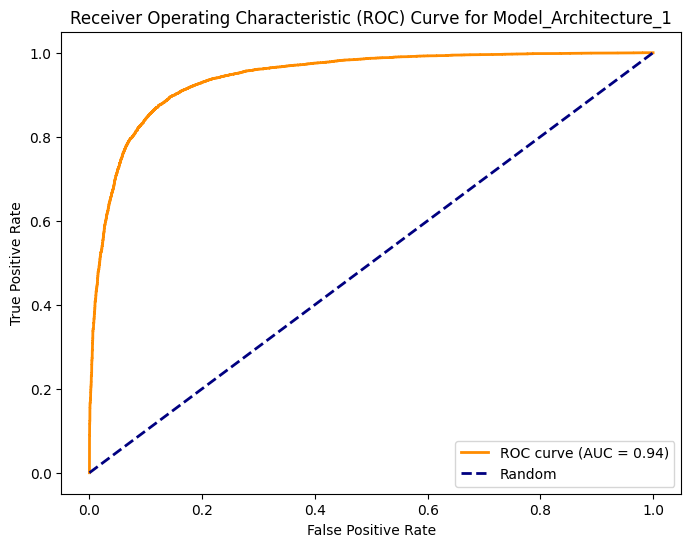
## Model 1 Architecture :

Simple architecture with a single Conv2D layer and minimal additional layers. Fast training and relatively good accuracy. No transfer learning; it’s built from scratch. Suitable for tasks where a simple model is sufficient, and computational resources are limited.

* Accuracy & Loss Predictions

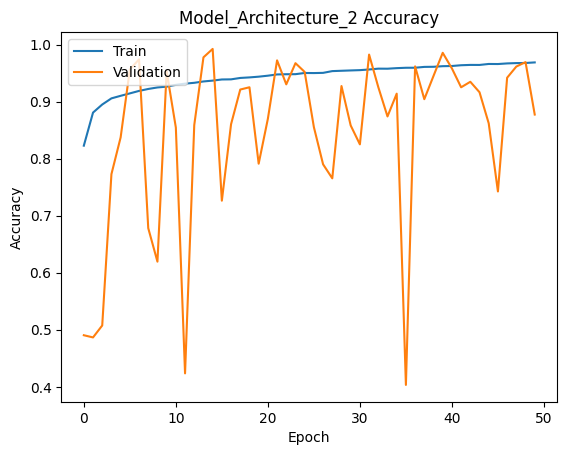
* Roc Prediction



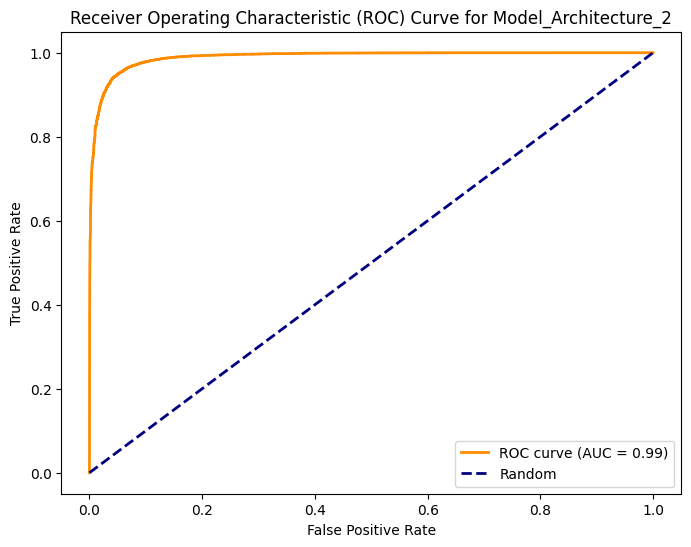
## Model 2 Architecture:

More complex architecture with multiple Conv2D layers and additional layers. Longer training time due to the increased complexity. Suitable for tasks where a more intricate model is needed, but it requires more time and resources for training

\*Accuracy & Loss Predictions

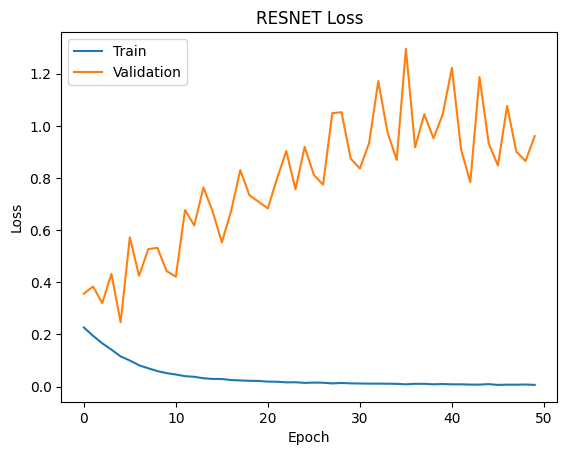
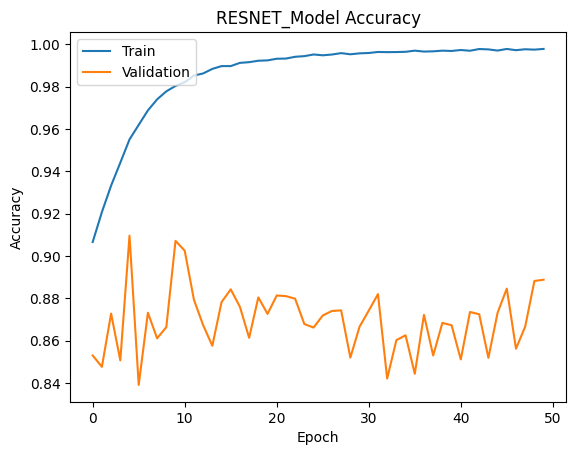
\*Roc Prediction



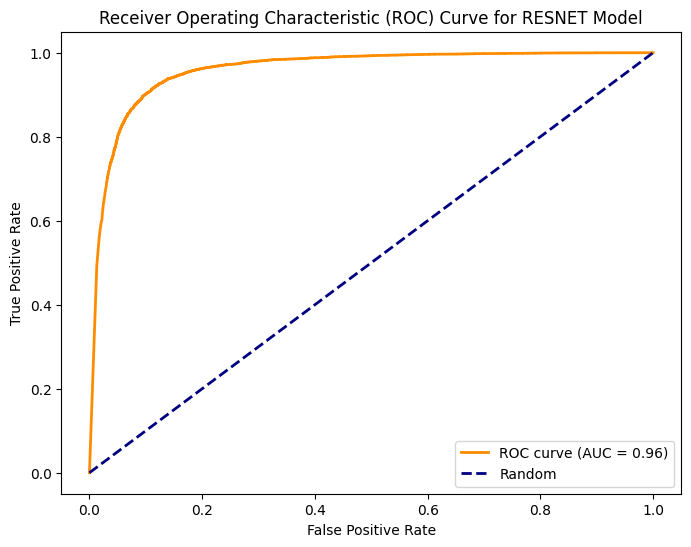
## Model 3 Resnet Architecture :

Uses transfer learning with ResNet50. Adds custom layers for the task. Freezes pre-trained layers, saving training time. Ideal for powerful models with pre-trained features, efficient training

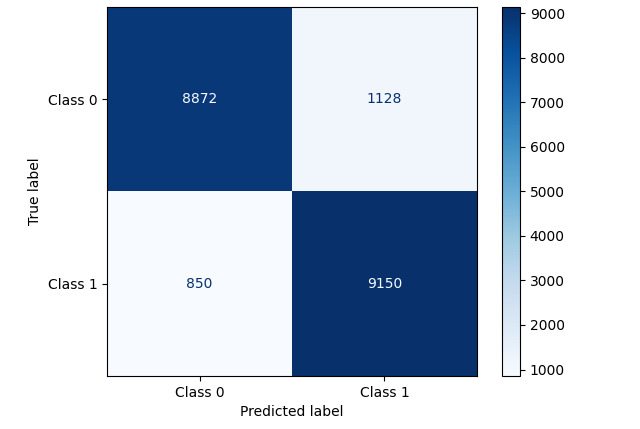
\*Accuracy & Loss Predictions

\*Roc Prediction



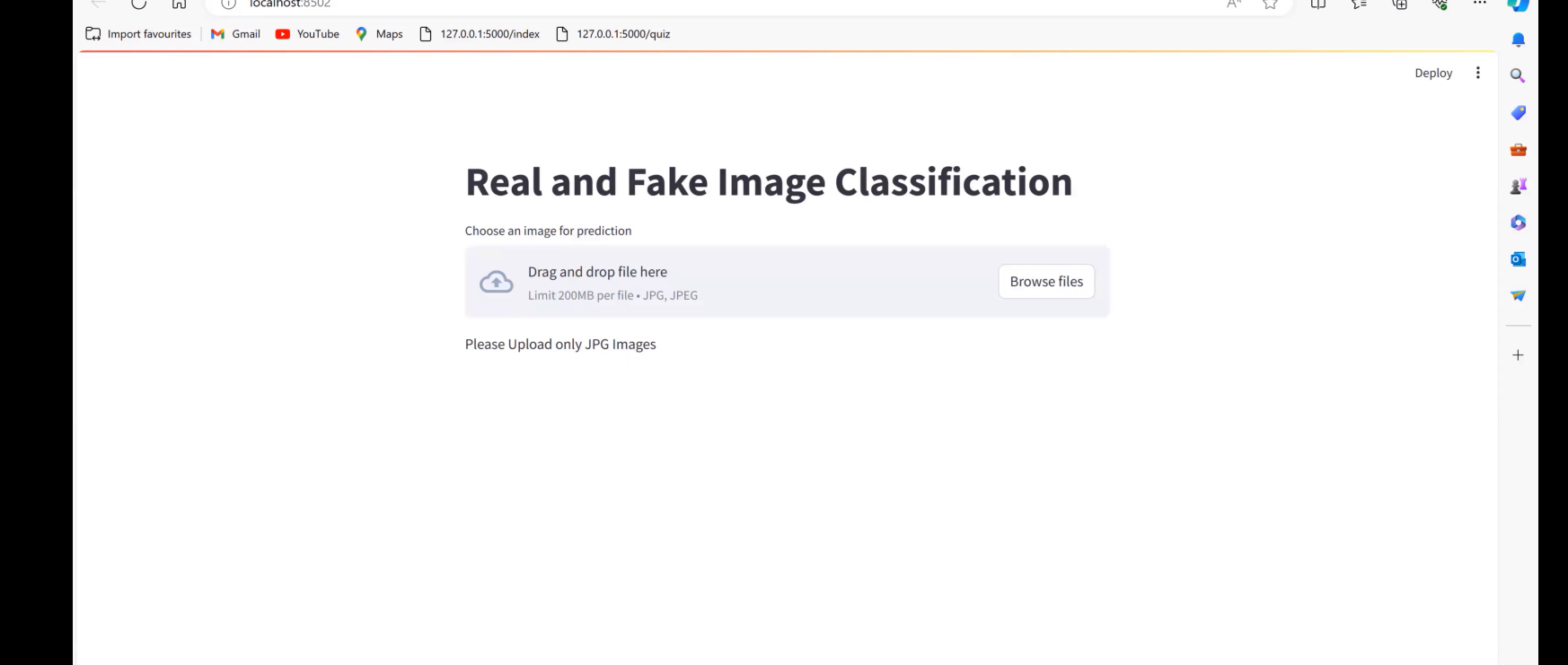
\*Confusion Matrix for the Best Model (RESNET50)



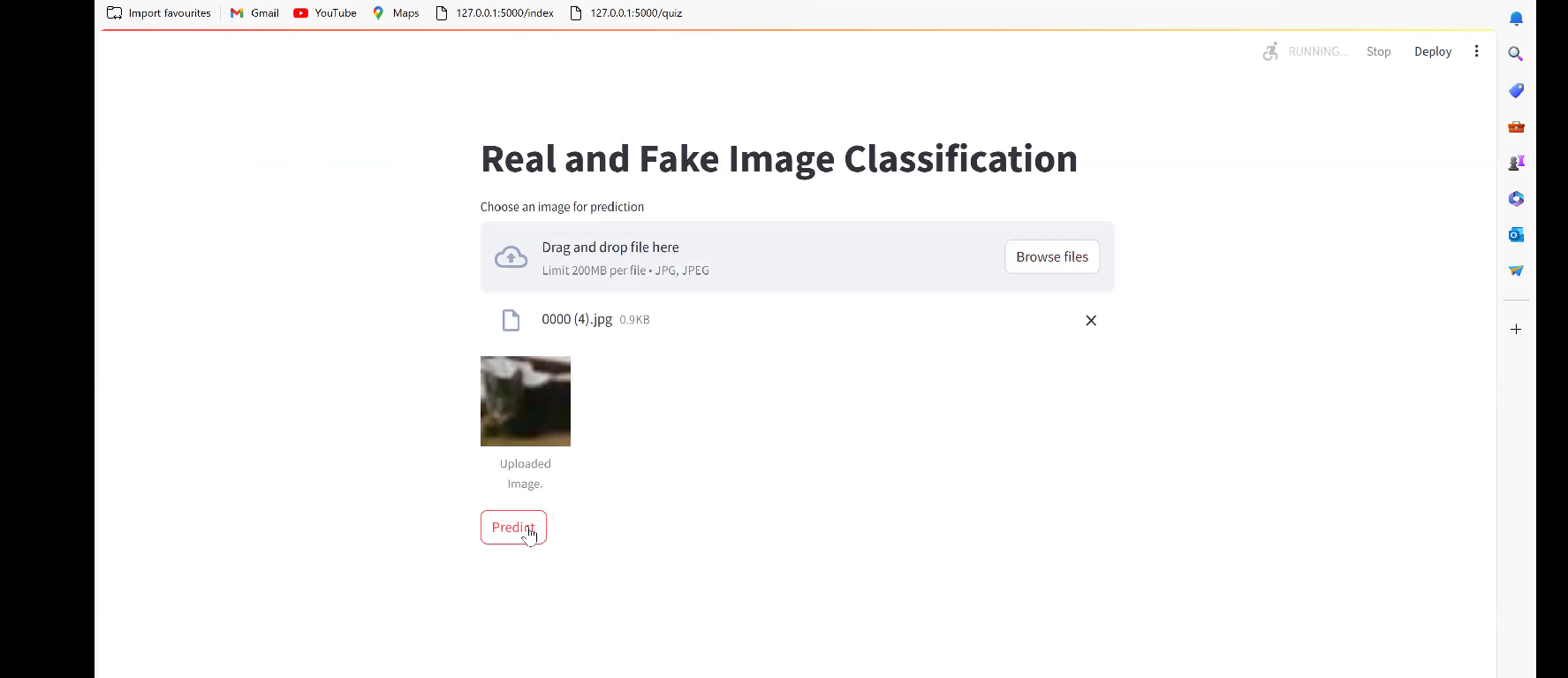
### 6. Application

The best and fastest model of these three has been saved in h5 format. Flask framework is used to create the backend. When the user opens the webpage, the index page is shown which allows the user to upload any image Once the user upload image and clicks on submit, the flask service listens to it and triggers a script This script loads the saved h5 model Preprocesses the input image Runs the model on the processed image And finally predicts the result The result is returned to frontend through which the user can see whether the image is real or fake fake.

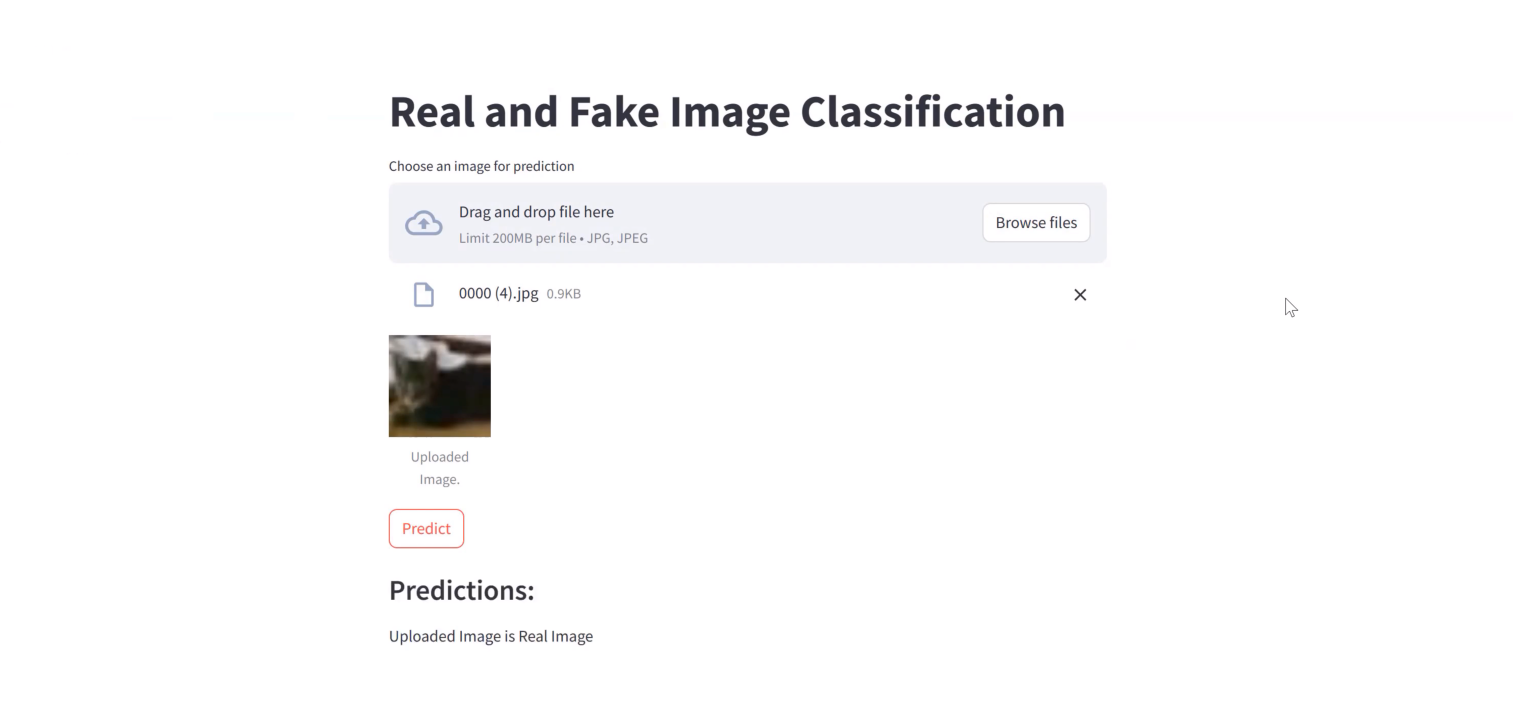
1. **web page created**



1. **Uploading and Predicting**



1. **Result**



### 

### 7. Conclusion

Atlast I was successfully able to apply machine learning approach to solve the task of image authenticity prediction. When it came to differentiating between actual and fake photos, the Convolutional Neural Network (CNN) models that were put into practice showed encouraging results. Important conclusions and highlights include:

Experimentation with Models: A variety of CNN architectures were investigated, taking into account various pooling layers and utilizing pre-trained models including VGG, ResNet, and Densenet. This made it possible to assess the model’s performance in great detail.

Data Augmentation: During the data preparation stage, random cropping, scaling, and rotating were among the image augmentation techniques used. This enhanced the models’ capacity to effectively generalize to a wide range of visual changes.

Performance Evaluation: Plots of loss visualization were used to assess the models’ performance on both training and validation datasets, and they offered valuable insights into the training procedure.The overall accuracy and robustness of the models were evaluated using the test dataset.

Outcome Analysis: A qualitative comprehension of the models’ performance was obtained by seeing sample images next to their actual and expected classes. A Confusion Matrix was utilized to further identify potential problem regions in the models.

Web application development: Gradio’s integration made it easier to create a web application that is easy to use. It’s simple for users to upload photos and get authenticity forecasts.

Future Research Work :

* Developments in Deepfake Detection:

Examine and create cutting-edge methods designed especially for identifying deepfake photos. This might entail investigating cutting-edge neural network topologies, making use of generative models, and deepening our comprehension of deepfake generating patterns.

* Multimodal Methods:

To improve the accuracy of fake picture identification, investigate the integration of several modalities, such as examining contextual information, captions, or metadata in addition to the image content.

* Reasonability in Identifying:

Improve the detection algorithms’ explainability to offer precise explanations for the classification of an image as real or fraudulent. Gaining user confidence and using this for forensics both depend on it.

* Transfer Knowledge Between Domains:

Examine transfer learning techniques to modify detection models for various contexts and domains. This can enhance the model’s capacity for generalization and efficacy over a range of datasets.

* Customized Adversarial Education :

Taking into account the dynamic nature of image modification techniques, develop approaches for adaptive adversarial training to strengthen the model’s resistance to evolving adversarial attacks.

* Systems for Real-Time Detection:

Prioritize the creation of systems for detecting phony images in real time that can function well in dynamic internet settings like social media.

References:

1. Bird, J.J., Lotfi, A. (2023). CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images. arXiv preprint arXiv:2303.14126.
2. K, A., K, A., N, D., & K, D. S. M. (2023). Detecting fake images using machine learning. International Journal of Research Publication and Reviews, 4(4), 2063–2069. https://doi.org/10.55248/gengpi.2023.4.4.35702
3. A. Kawabe, R. Haga, Y. Tomioka, Y. Okuyama and J. Shin, “Fake Image Detection Using An Ensemble of CNN Models Specialized For Individual Face Parts,” 2022 IEEE 15th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC), Penang, Malaysia, 2022, pp. 72-77, doi: 10.1109/MCSoC57363.2022.00021.
4. Indolia, S., Goswami, A. K., Mishra, S., & Asopa, P. (2018). Conceptual understanding of Convolutional Neural Network- a deep learning approach. Procedia Computer Science, 132, 679–688. https://doi.org/10.1016/j.procs.2018.05.069