Indian Institute of Information Technology Allahabad

Prayagraj (UP) India

Identification of Artificially Generated Images using CNN

Supervised By :
Dr. Shiv Ram Dubey

Submitted by:
Shubham Kumar Bhokta
IIT2020007

Introduction

The rise of AI in image processing and computer vision has given rise to convincingly deceptive synthetic images.

These artificial images, which are frequently created using deep learning models, present a serious threat to the reliability and authenticity of visual content.

So we present a comprehensive approach to recognize artificially generated image using a ResNet-based framework.

LITERATURE REVIEW

[1] **Exposing computer generated images by using deep convolutional neural networks**

This paper uses a deep architecture based on a convolutional neural network (CNN) to classify each image from the dataset. The raw RGB pixels are used as dataset. The deep CNN architecture is based on the ResNet-50 model and the method uses transfer learning techniques.

[2] **Distinguish computer generated digital images: A CNN solution - Ming He**

The research utilises a Convolutional Neural Network (CNN) model that incorporates VGG19 and ResNet50 architectures. To tackle the challenge posed by limited training data, the study introduces the concept of transfer learning. The chosen loss function is Softmax, and Training Settings including data augmentation are discussed.

[3] **CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images - Jordan J. Bird, Ahmad Lotfi**

The study compares AI-generated images using synthetic data that looks like CIFAR-10. With a 92.98% accuracy rate, a Convolutional Neural Network (CNN) classifies images as real or artificial intelligence (AI) generated.

Dataset Description

Our dataset comprises both manipulated and authentic images, with the former depicting artificially generated faces through various methods.

Each image in the dataset is a 224 x 224 jpg representation of a human face, presenting either a genuine or fabricated portrayal.

Subset	#Images	#Faces	#Real Faces	#Forged Faces
Training	44,122	151,364	85,392	65,972
Validation	7,308	15,352	4,786	10,566
Test-Development	18,895	49,750	21,071	28,670
Test-Challenge	45,000	117,670	49,218	68,452
Total	115,325	334,136	160,670	173,660

ResNet(Residual Neural Network)

1) **Residual Block**: Introduces a building block that enables the learning of residual functions, addressing the vanishing gradient problem in deep neural networks.

2) **Skip Connections**: Connects input and output across layers, facilitating the flow of information and mitigating degradation issues in the training of deep networks.

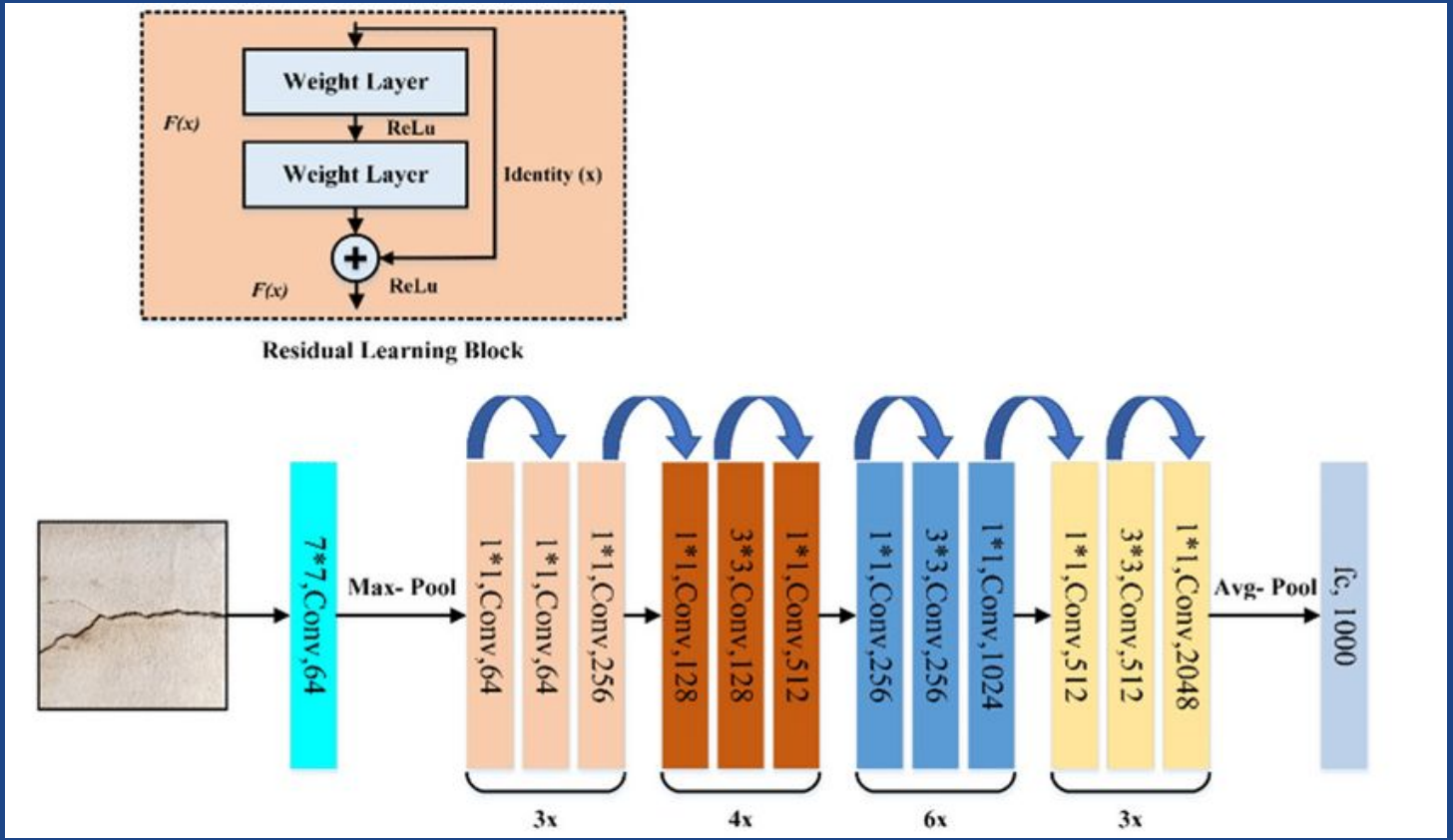
3) **Identity Mapping**: Aims to learn an identity function within residual blocks, enhancing model convergence and enabling the training of deeper networks.

4) **Bottleneck Architecture**: Utilizes a three-layer structure in residual blocks, reducing computational complexity and enhancing the efficiency of deep neural networks.

5) **Architecture Depth:** ResNet achieves remarkable depth, enabling the training of very deep networks with hundreds or even thousands of layers.

6) **Batch Normalization**: Normalizes intermediate layer inputs, reducing internal covariate shift and accelerating training convergence in deep neural networks.

Architecture



Experimental Settings

In our settings, we configured the notebook options as follows:

- **Accelerator**: GPU T4*2

- **Language**: Python

Methodology

1. Data Preparation

- Organized dataset into training, validation, and test sets with real and fake images.
- Ensured balanced distribution in each dataset split using the `check_dist()` function.

2. Data Preprocessing

- Resized images to 224x224 pixels (IMG_SIZE).
- Applied data augmentation techniques, including rescaling, using ImageDataGenerator.

3. Model Building

- Constructed a CNN model using transfer learning with ResNet50 as the base model.
- Froze base model layers except for the last 150 layers to prevent overfitting and retain pre-trained weights.
- Added global average pooling, dropout, and dense layers for feature extraction and classification.
- Compiled the model with Adam optimizer, categorical cross-entropy loss function, and accuracy metric.

Methodology

4. Model Training

- Trained the model on training data (train_flow) for 15 epochs, validated on validation data (valid_flow).
- Used ModelCheckpoint and Early Stopping callbacks to save the best model and prevent overfitting.

5. Model Evaluation

- Evaluated the trained model on test data (test_flow) to obtain test loss and accuracy.
- Computed additional metrics like confusion matrix and classification report using scikit-learn.

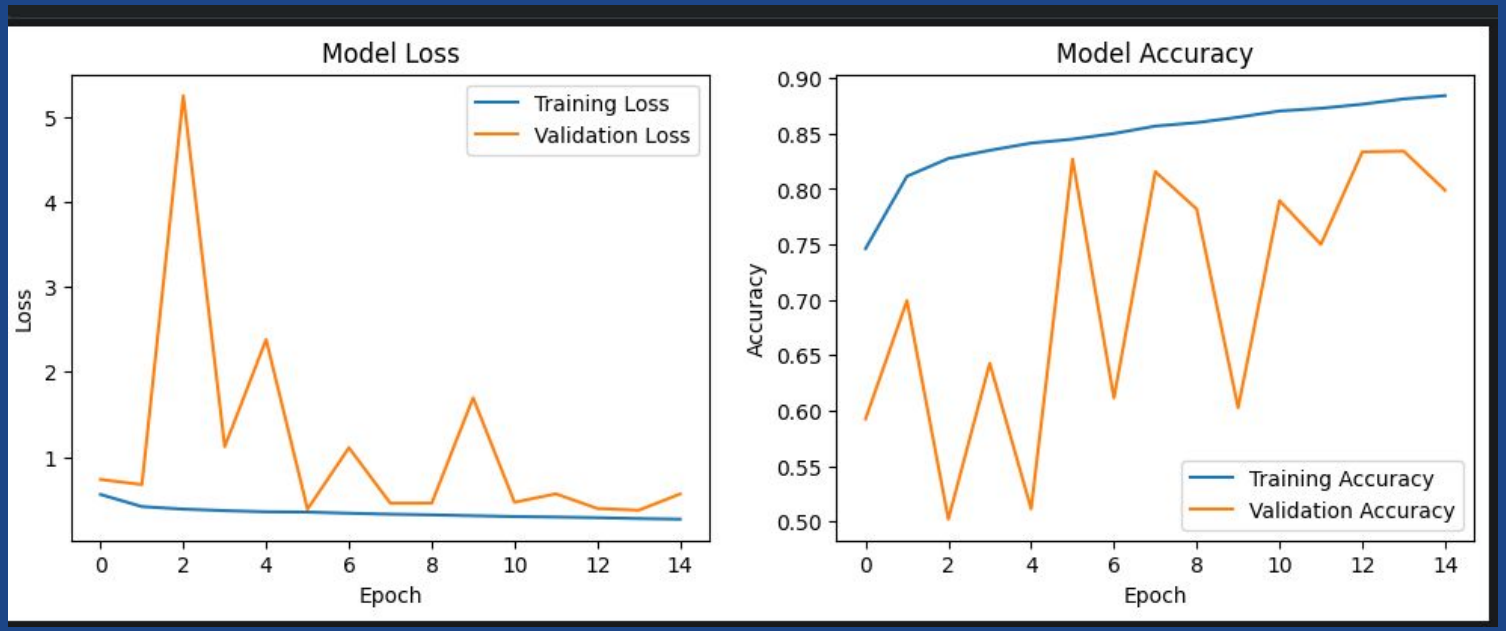
6. Reporting

- Visualized training history (loss and accuracy over epochs) using Matplotlib.
- Displayed individual image predictions with actual and predicted labels for qualitative analysis.

7. Testing

- Assessed the fine-tuned model's generalization performance on the held-out testing set.
- Achieved 77% accuracy, 75% recall, and 74% F1 score on the testing set.

Results and Analysis



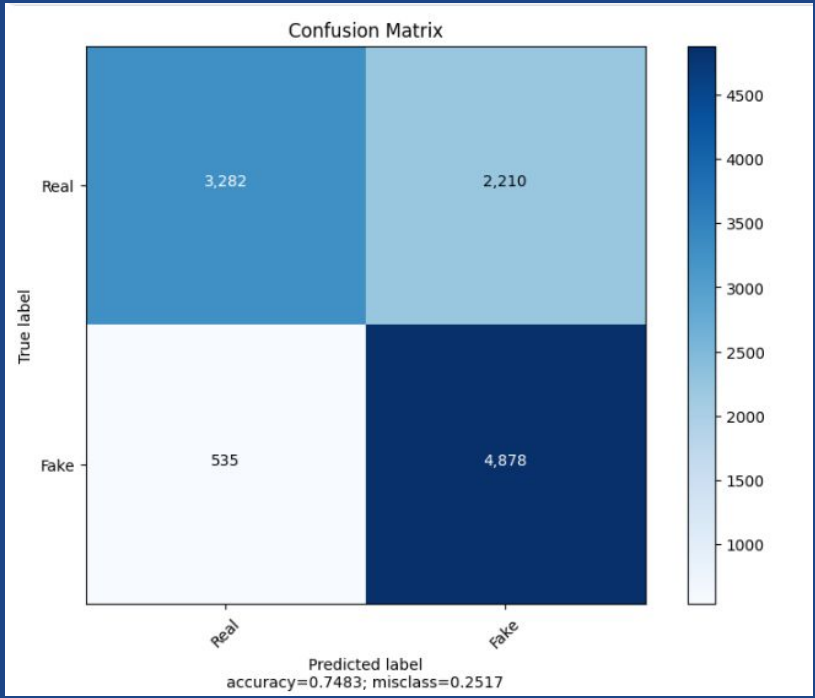
Results and Analysis

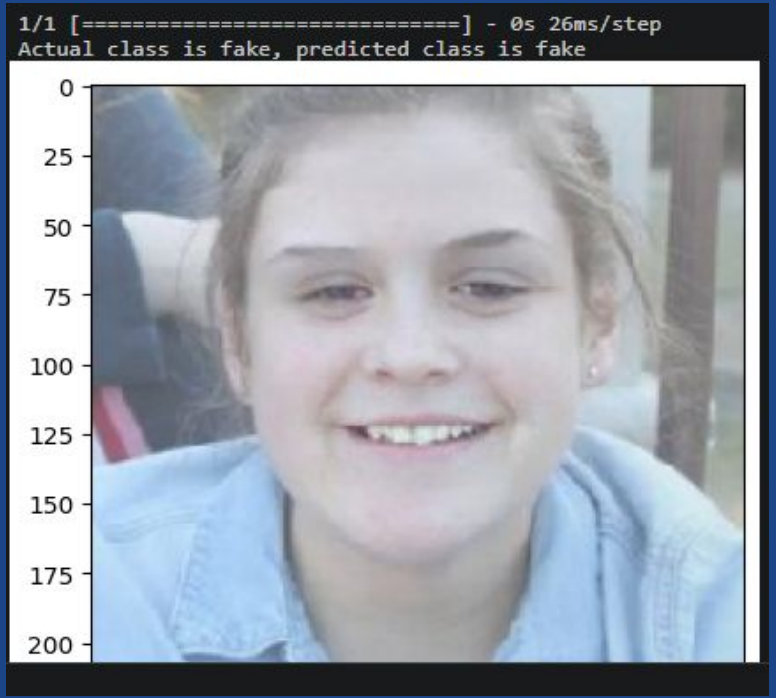
- The accuracy of our model stands at a noteworthy 77%, showcasing its proficiency in predictions.
- With a precision of 86%, our model excels in correctly identifying positive instances with accuracy.
- The F1-score, a balanced metric at 0.71, underscores our model's effectiveness in minimizing false predictions.

Results and Analysis

```
10905/10905 [==============================] - 88s 8ms/step
Confusion Matrix:
[[3282 2210]
 [ 535 4878]]

Classification Report:
              precision    recall  f1-score   support

        Fake       0.86      0.60      0.71      5492
        Real       0.69      0.90      0.78      5413

    accuracy                           0.75     10905
   macro avg       0.77      0.75      0.74     10905
weighted avg       0.77      0.75      0.74     10905
```

Results and Analysis





Conclusion

Because AI and deep learning are developing so quickly in the digital age, it is more important than ever to identify artificially generated images.

These technologies enable the creation of extremely lifelike counterfeit images, opening the door to potential abuse.

To tackle this obstacle, a variety of inventive methods—from conventional forensics to state-of-the-art deep learning are needed.

And our method serves as an efficient way to identify Artificially Generated Images.

References

1. Bird, J. J., & Lotfi, A. (2023). CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images. arXiv preprint arXiv:2303.14126 [cs.CV].Paper Link

2. Khoo, B., Phan, R. C.-W., & Lim, C. H. (2021). Deepfake attribution: On the source identification of artificially generated images. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 12*(7). DOI: 10.1002/widm.1438.Paper Link

3. Gragnaniello, D., Marra, F., & Verdoliva, L. (2022). Detection of AI-Generated Synthetic Faces. In Handbook of Digital Face Manipulation and Detection (pp. 191–212). Advances in Computer Vision and Pattern Recognition ((ACVPR)). Open Access. Paper Link

Thank You