CS 59000 Application of Deep Learning

Homework 6: Detecting Real and AI-Generated Synthetic Images

Due on **November 6**, 2024, 11:59pm

The goal of this homework assignment is to help you become familiar with convolutional neural networks (CNNs) and apply a Hugging Face model for image classification. For Parts 1 and 2, please include all code in the same Jupyter Notebook, named "parts_1_2.ipynb". Ensure that all outputs are retained in the Jupyter Notebooks.

This assignment uses the CIFAKE dataset, which contains real and AI-generated synthetic images, available on Kaggle:

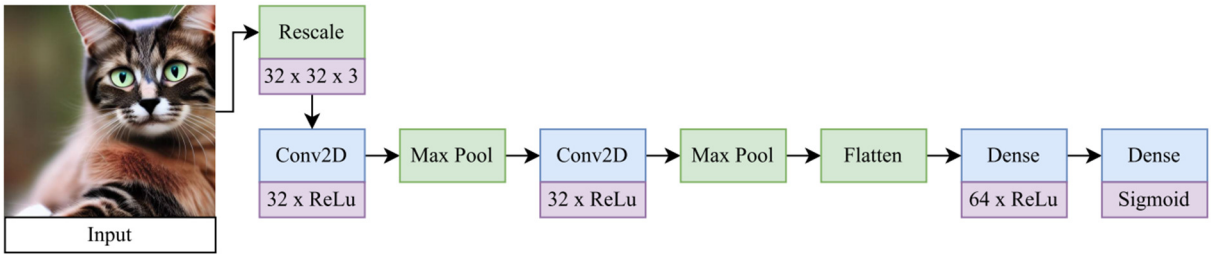
https://www.kaggle.com/datasets/birdy654/cifake-real-and-ai-generated-synthetic-images

The dataset includes 100,000 training images (50,000 per class) and 20,000 testing images (10,000 per class). Download the dataset (both training and testing) as a zip file and upload it to your Google Drive.

1. **Split the training dataset into a real training dataset and a validation dataset**. In the Jupyter Notebook "parts_1_2.ipynb", write Python code to accomplish the following:
   - Connect to your Google Drive and navigate to the folder containing the zip file.
   - Unzip the file (this process may take about 30 minutes). After unzipping, you will find two folders: "train" and "test".
   - Create a third folder named "validation". Move 10,000 images from "train/REAL" to "validation/REAL" and 10,000 images from "train/FAKE" to "validation/FAKE".

   As a result, the train folder will contain 80,000 images, while the validation folder will have 20,000 images.
2. **Convolutional Neural Network (CNN)**: Use the image_dataset_from_directory() function to load data from the "train", "validation", and "test" folders. Build the CNN model as described in the research paper "CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images" (IEEE Access) at

   https://ieeexplore.ieee.org/abstract/document/10409290, i.e.,



which is the Figure 5 in the paper for the model architecture. Apply the “ModelCheckpoint” callback to save the best model based on validation loss during training, and train the model for 30 epochs. Ensure that your final model achieves at least 92% accuracy on the test dataset. Note that it may take several hours (e.g., 3 hours) to train the model.

3. **SDXL Detector from Hugging Face**. Access the SDXL detector on Hugging Face at:

   https://huggingface.co/Organika/sdxl-detector

   Use this model to evaluate the accuracy of detecting both real and AI-generated images in the **test** dataset. Create a new Jupyter Notebook named “part_3.ipynb”, and write Python code to perform the following tasks:

   - Connect to your Google Drive and navigate to the folder containing the test dataset.
   - Apply the SDXL detector to all images in the “test/REAL” folder and calculate the detection accuracy, defined as the percentage of images correctly classified as the "human" class.
   - Apply the SDXL detector to all images in the “test/FAKE” folder and calculate the detection accuracy, defined as the percentage of images correctly classified as the "artificial" class.

4. **Bonus**. Apply a different model, either one you implement and train yourself or an existing pre-trained model. Bonus points will be awarded based on the model’s accuracy on the test dataset as follows:
   - 93 ≤ Accuracy < 94: 20pts
   - 94 ≤ Accuracy < 95: 35pts
   - Accuracy ≥ 95: 50pts

   Note: The test dataset must not be used during training or validation. Put your implementation into “part_4.ipynb” file with all outputs retained.

Please upload “parts_1_2.ipynb” and “part_3.ipynb” files to your GitHub homework repo under “hw6”. If you work on part 4, please also upload the “part_4.ipynb” file for the bonus points.

Grading rubric:

- Part 1 – 20pts
- Part 2 – 50pts
- Part 3 – 50pts

Total: 120pts