**COMP4423 – Assignment 2**

**LEUNG Kit Chuen (20051248D)**

1. **Task Requirement**

It is a **classification task** because the goal is to classify images into two categories: real-world pictures captured on the PolyU campus and AI-generated images of corresponding architectures. In other words, the task involves determining whether an image is genuine or artificially generated. This falls under the realm of classification problems, where the aim is to assign a label or category to input data based on its features.

In this scenario, a Convolutional Neural Network (CNN) model can be trained to learn the distinguishing features between real-world images and AI-generated images. The model will analyze the patterns, textures, and structures present in the images to make predictions about their authenticity. In fact, studies have shown that AI-generated images have explainable features which is varied from real images [1]. Therefore, it is confident for using CNN to complete the tasks.

In addition, we are encouraged to classify the false images into different art styles. Using a classification model remains an effective solution to solve the problem. Therefore, the required CV task shall be constructing a model which classifies the real pictures and the false pictures with different visuals simultaneously.

1. **Data Collection and Labeling**

It is assigned that data collection shall be performed manually. During collection, a total of 174 real images were collected. These images are taken in various venues, such as the platform lawn, the Innovation Tower, and the Z core platform. In addition, a total number of 94 AI-generated pictures are created with getimg.ai [2]. The AI tool provides three art styles: photorealism, artistic, and anime. There are 33, 33, and 28 fake images for each corresponding style.

For validating the model, 70% of the images are used as training set while 30% are used as validation set. After reaching the satisfying score, we are using a separate test dataset to validate the accuracy.

Originally, we attempted to fill the class with string labels. However, it raised errors and we are forced to change the labels into numeric, and use One Hot Encoder to further modify them.

1. **Image Preprocessing**

Image preprocessing is a crucial step in computer vision projects. In the context of classifying real and AI-generated images of the PolyU campus, the following preprocessing steps are recommended:

* **Resizing**: Images in the dataset may vary in size, which can introduce inconsistency and computational overhead during training. Resizing all images to a uniform resolution not only ensures consistency but also reduces computational complexity. The resizing operation should maintain the aspect ratio of the original images to prevent distortion. The size shall depends on the input of each model.
* **RGB Standardization**: Standardizing the color channels of the images is essential for ensuring consistency across the dataset. This involves normalizing the pixel values in each color channel (red, green, and blue) to have zero mean and unit variance. Standardization helps in stabilizing the learning process and improving convergence during training.

In addition, to enhance the robustness of the models, data augmentation is applied to the training and validation set. The provided function *augmentImage* is an image augmentation function designed to achieve the goal, particularly those trained on image data. Image augmentation is a common technique used in computer vision tasks to artificially increase the diversity of the training dataset by applying random transformations to the input images. This helps the model generalize better to real scenarios and reduces overfitting.

Let's break down the different types of image augmentation techniques applied in the function:

* **Rotation**: Randomly rotates the image between -45 and 45 degrees. This helps the model learn to recognize objects from different viewpoints and orientations.
* **Flip**: Randomly flips the image horizontally or vertically. This introduces variations in the orientation of objects within the image.
* **Scaling**: Randomly scales the image between 0.8 and 1.2 times its original size. This simulates changes in the distance between the camera and the object, as well as changes in object size.
* **Translation**: Randomly translates the image by up to 20 pixels in both horizontal and vertical directions. This simulates shifts in the position of objects within the image.

Each augmentation operation is applied based on the value of *random\_seed*. By using the modulo operation with different prime numbers (2, 3, 5, 7), the function ensures that each augmentation operation has a different probability of being applied. This helps introduce a diverse range of transformations to the images while avoiding over-augmentation.

1. **Model Selection**

Firstly, we shall focus on the real vs fake classification task. We are going to test various of CNN classification models, including AlexNet,

1. **AlexNet**

AlexNet is one of the earliest CNN models, proposed in 2012. Its architectural innovations and training strategies paved the way for the development of deeper and more powerful CNN models, revolutionizing the field of computer vision and contributing to significant advancements in image classification tasks. To begin with, we need to reconstruct the AlexNet, with Tensorflow Karas. It was relatively easy to reconstruct as the detail of the model is already given. The following table shows the structure of AlexNet.

一張含有 文字, 螢幕擷取畫面, 功能表, 字型 的圖片

自動產生的描述

1. **References**

[1] J.J. Bird, A. Lotfi, "CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images," arXiv, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2303.14126

[2] Webrockets, 2024. *getimg.ai* (Version 1.0) [Image Generator]. [Online]. Available: https://getimg.ai/features/image-to-image

**1. Introduction**

This is a template for the COMP4423 Assignment 2 report.

Your report should be coherent with the code and the result, otherwise, you will lose marks.

**2. Method**

In your report, these questions should be answered:

1. In order to implement this system, what CV task do you think of it as?
2. What algorithms and models do you use?
3. How do you train and test the model?
4. How do you design to ensure the robustness of the model in a real scenario (color and scale variations)?
5. What problems do you find and how do you solve them?

**3. Structure of your report**

There should be a section to discuss the above questions. Except for this, there is no requirement for the report.