MAIS202 – Deliverable 2

Problem Statement

Our project proposes to colourize black and white photos.

Data Preprocessing

Our dataset will be from the following Kaggle:

1) https://www.kaggle.com/datasets/shravankumar9892/image-colorization

This dataset contains 25000 images in the Lab colour space represented as NumPy arrays. Each image was resized to be 224 x 224 pixels in dimension. The training input (L) and training output (ab) vectors were extracted and stored in separate arrays. For future reference, should we need to expand the dataset, these parameters (colour space and dimension) will be kept constant.

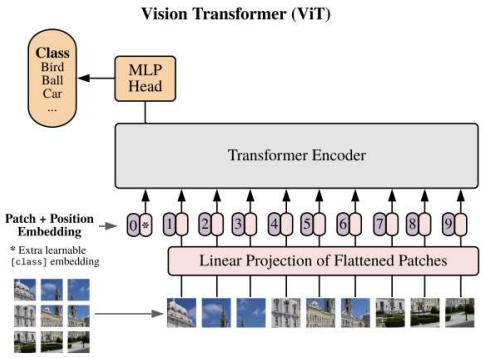
We split the training data (25,000) into sets of 20,000 for training, 2,500 for testing, and a further 2,500 for validation.

The *VitMAEForPreTraining* class provided by the *transformers* library takes in an image input with three channels. Since our input only has one channel (greyscale), we can duplicate these channels to create three identical channels that can be read by the pre-trained model (introduced below).

Machine Learning Model

Our chosen model was changed to a Vision Transformer (*ViT*). Vision transformers are derived from the transformer model architecture, which consists of an encoder and decoder. What separates the two is that vision transformers are applied over patches of input images. These patches are then flattened and linearly embedded to be used as input to the standard transformer encoder [1].

The main advantage of using a *ViT* over a *CNN* is that less information is lost as the input is not reduced in dimensionality through pooling, giving more accurate results.



The model was implemented using a pre-trained model from the Hugging Face hub. The two models of interest are *ViTModel*, which is the bare *ViT* model transformer, and *ViTMAEModel*, which is a *ViTModel* specifically trained to reconstruct masked images.

- Framework:
    - PyTorch or Jax
    - Transformers (from hugging face)

        - *ViTMAEForPreTraining*
            - Includes decoder, trained specifically for reconstructing masked patches
        - *ViTModel*
            - Only outputs the hidden states at the last layer of the model, the forward pass needs to be defined (decoder not included? would have to train ourselves)
- Other tools:
    - *NumPy*
    - *PIL*
    - *matplotlib*

Options:

1) *ViTMAEForPreTraining*
    a. It's possible we can override the configs and set the *mask_ratio* to zero, then just fine-tune the model from there
2) *ViTModel*
    a. We create our own decoder, will probably take more time

The main challenge for this stage was learning how to use Hugging Face and understanding the basics of *ViTs* as this was not a model presented in the course.

Preliminary Results

For this stage of the project, we explored the Hugging Face hub. The focus was to learn how to import a pre-trained model and use it to output an image. This was done by "*unpatchifying*" the model's output logits using the *PIL* and *torch* libraries [2].

Next Steps

- https://github.com/huggingface/transformers/tree/main/examples/pytorch/image-pretraining
    - Tutorial on how to pre-train ViTMAEForPreTraining for our own purposes
- https://huggingface.co/docs/transformers/training
    - Or this
- Fine tuning the model
    - The trainer class provided within the transformers package allows us to train our ViT model without the need or creating our own training function/loop. The trainer class will take training parameters that we will need to specify. We can use the MSE metric within the evaluate library to compare the predictions generated by our model against the values of the coloured images. We will have to create a function to evaluate our predictions, using the previously mentioned library and pass that through to the trainer class. Training, then, should simply be instantiating the trainer class with the model, parameters, dataset, and evaluation function passed and calling the train function.

References

[1] Papers with Code - Vision Transformer Explained. (n.d.). https://paperswithcode.com/method/vision-transformer

[2] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, & Ross Girshick. (2021). Masked Autoencoders Are Scalable Vision Learners. ArXiv: Computer Vision and Pattern Recognition.