

Interim Report

Project Overview

This project will focus on researching and evaluating the performance of EfficientNet for image generation tasks in deep learning. This will be done primarily with the models' integration into a CycleGAN, but will also be incorporated in some other, more simple networks. The performance will be evaluated by comparing against data recorded in the official papers, as well as through the images output by my own models.

Summary of work to date

Network research – Deciding on the best network for swapping images of Pepsi cans to Coke cans. Discovered image translation/generation models which included InstaGAN, CycleGAN, StyleGAN.

Blog post - <https://deepads.ai/2020/09/03/network-research/>

Choosing the right network – After considering multiple networks used for image generation and translation, the CycleGAN was selected as the focus of the project. Blog post -

<https://deepads.ai/2020/09/06/cyclegan/>

Created a custom dataset – Gathered hundreds of Coke and Pepsi can images online and saved into respective folders. Created a python script that imports images from a folder, appends to list and converts to NumPy array. The NumPy array is then saved as a .npz file for use in the CycleGAN model.

CreateDataset python file - <https://github.com/Chris-Mayes/Dissertation/blob/main/model/createdataset.py>

blog post - <https://deepads.ai/2020/09/06/creating-the-dataset/>

Pre-Processing – The datasets created in the previous stage were imported to the CycleGAN file and used in creation of TensorFlow datasets. `Tf.data.Dataset.from_tensor_slices()` was used for this. They were then normalized and pre-processed with random flips and crops.

Cells 2-8 of initial commit - <https://github.com/Chris-Mayes/Dissertation/blob/main/model/model.ipynb>

Setting up basic CycleGAN model – A first iteration of the CycleGAN was created using the guidance from the CycleGAN paper and TensorFlow implementation.

Paper - <https://arxiv.org/abs/1703.10593>,

TensorFlow model guide - <https://hardikbansal.github.io/CycleGANBlog/>,

GitHub code - <https://github.com/Chris-Mayes/Dissertation/blob/main/model/model.ipynb>

Researching DAEs and EfficientNet – Contemplated the integration of a denoising auto encoder but decided to focus on having a working CycleGAN with EfficientNet first.

Blog post - <https://deepads.ai/2020/11/12/research-update/>

Implemented EfficientNet – Replaced ResNet generator with custom EfficientNet based generator. Had to learn more about transfer learning. Used entire EfficientNet model with modified output

layers to upscale. Was not performing quite as well as ResNet yet.

GitHub image - [https://github.com/Chris-](https://github.com/Chris-Mayes/Dissertation/blob/devChris/documentation/Training%20outputs/efficient1.png)

[Mayes/Dissertation/blob/devChris/documentation/Training%20outputs/efficient1.png](https://github.com/Chris-Mayes/Dissertation/blob/devChris/documentation/Training%20outputs/efficient1.png)

Updated EfficientNet – Changed how EfficientNet was being used. Only a small portion of the layers needed to be included as the down sampling and upscaling aspect of the original CycleGAN was important.

Blog post - <https://deepads.ai/2020/12/09/model-update/> (image of better output included here)

Continued work to increase model performance – Changed direction of the project to focus on the performance of EfficientNet in image generation. Added loss tracking and training checkpoints to work out where the model could be underperforming.

Blog post - <https://deepads.ai/2021/01/01/december-jan-update/>

Evaluation

All milestones up to the current date have been mostly achieved, however the plan for the project has changed over time. By December I had planned to have fully implemented EfficientNet, though after many weeks of research and work on the model, it was clear that this would be a much harder task than originally anticipated. Through this discovery, it was decided that the project direction would change from a system for advertising to a research-based project evaluating the performance of EfficientNet for image generation. This change in direction occurred as a result of being unable to find the necessary resources online to aid my original project. The current guides and resources related to EfficientNet only cover image classification and it is evident that no one has explored and reported on its use for generation, possibly because it is a new model, only having been published in 2019. Due to it being an unexplored topic, it proved to be a clearer path to guide future work. The original scope was not completely unrealistic but after facing the challenges with EfficientNet, it was clear that the original timeframes would have been hard to meet. It is still undecided if EfficientNet will provide good enough results in the CycleGAN and produce high quality images, good enough for advertising, but this current uncertainty is why I believe it makes a good research project. All aspects of the project up to the implementation of EfficientNet produced expected results or better. Even with the small custom dataset of coke and Pepsi cans, the early iteration of the network learned a mapping between the two. Due to the lack of online documentation, I had no way of setting an expected baseline for the performance of the model with EfficientNet, however, the current stage of the project shows promising results. Through the workflow up to now, I have learnt a lot about image generation and translation models, transfer learning, dataset creation and pre-processing, using functional models, and tracking and monitoring loss.

Revised Project Plan

The primary focus for the remainder of the project is to gather as much data about the performance of EfficientNet for image generation as possible. This will include continued work on the CycleGAN model to achieve the best results in the remaining time, as well as incorporating the EfficientNet model in other GANs.

Timeline and Milestones

Date	Milestone	Completed	Allocated Time
09/11/2020	Finalise Project idea and have good understanding of MVP	Yes	1 week
16/11/2020	Research CycleGAN algorithm	Yes	1 week
23/11/2020	Compile larger dataset		1 week
30/11/2020	Complete CycleGAN base model	Yes	2 weeks
07/12/2020			
14/12/2020	Research and implement EfficientNet architecture in CycleGAN	Yes	2 weeks
21/12/2020			
28/12/2020	Record losses to compare modifications in code	Yes	1 week
04/01/2021	Add checkpoints for saving the model Interim Report	Yes	2 weeks
11/01/2021	Interim report due Test outputs at intermediate layers		2 weeks
18/01/2021			
25/01/2021	Continue tweaking model		1 week
01/02/2021	Test Model and make changes as necessary		2 weeks
08/02/2021	Begin working on draft submission		8 weeks
15/02/2021	Ensure CycleGAN results are recorded for analysis		1 week
22/02/2021	Research models to implement EfficientNet		1 week
01/03/2021	Set up and test EfficientNet in DCGAN/other networks		1 week
08/03/2021	Tweaking model to obtain best output		1 weeks
15/03/2021	Project Report		9 weeks
22/03/2021			
26/03/2021	Draft report submission due		
14/05/2021	Final report is due		

Currently researching the following

Testing outputs - https://keras.io/getting_started/faq/#how-can-i-obtain-the-output-of-an-intermediate-layer-feature-extraction

EfficientNet - <https://medium.com/@nainaakash012/efficientnet-rethinking-model-scaling-for-convolutional-neural-networks-92941c5bfb95>

DCGAN - <https://www.tensorflow.org/tutorials/generative/dcgan>

Metrics - https://keras.io/api/metrics/probabilistic_metrics/#binarycrossentropy-class

Binarycrossentropy for generators - <https://stats.stackexchange.com/questions/242907/why-use-binary-cross-entropy-for-generator-in-adversarial-networks>

Repo - <https://github.com/Chris-Mayes/Dissertation/tree/devChris>