Life Could Be a Dream: Somnial Units for Binary Recyclability Classification

Du Bowei, Gregory Lim, Ryan Cheong
Singapore University of Technology and Design
Singapore, SG 487372
{bowei_du,gregory_lim,ryan_cheong}@mymail.sutd.edu.sg

Abstract

We propose a novel architectural unit we call a *somnial unit*, and demonstrate its use on a waste recyclability task unique to Singapore.

1 Introduction

The separation of household waste is essential for effective recycling, yet Singapore continues to face a significant issue of recycling contamination: non-recyclable items being incorrectly disposed of in recycling bins. Manual sorting of waste is both labor-intensive and costly, posing a challenge for scalability. Waste recyclability classification can be defined as the process of determining whether an image of waste corresponds to a nonrecyclable or recyclable object. In this paper, we introduce a novel architectural unit we call a *somnial unit*, and demonstrate its use on a waste recyclability task unique to Singapore.

This project is available at

`http://github.com/BoweiDu01/50021AIGrp21`

where instructions to reproduce our results can be found in the `readme.md` file.

2 Related work

In 2021, Zheng et al. introduced EnCNN-UPMWS, an ensemble model that combines predictions from GoogLeNet, ResNet-50, and MobileNetV2 to classify waste images more accurately [2]. While the literature addresses waste classification based on material types, limited research has been conducted on local adaptations of the task to specific national recycling standards, such as those outlined by the National Environment Agency (NEA) in Singapore. This gap in the literature provides the motivation and novelty for our work.[1]

3 Proposed unit

To support our goal of accurate and efficient waste recyclability classification, we propose a novel convolutional architectural unit inspired by neurobiological notions of dreaming in animals.

Let $C \in \mathbb{N}_{\geq 1}$ be an image channel count, and $H \in \mathbb{N}_{\geq 1}$ be an image height, and $W \in \mathbb{N}_{\geq 1}$ be an image width, and $L \in \mathbb{N}$ be a memory buffer size, and rand denote a uniform pseudorandom selection function.

Definition 1 (Memory Buffer)**.**

$$\mathcal{M}_t = \{\mathsf{x}_s : \max(0, t - L) \leq s \leq t\}$$

[1] We make no direct comparison to any existing model in this report as we could not find a fair, "apples-to-apples" comparison, given our task.

Definition 2 (Recollector)**.** Let $\mathbf{W}_\rho \in \mathbb{R}^{1\times 1\times C\times C}$ be a recollector weight, and $\boldsymbol{b}_\rho \in \mathbb{R}^C$ be a recollector bias.

$$\rho : \mathbb{R}^{H\times W\times C} \to \mathbb{R}^{H\times W\times C}$$
$$\rho(\mathsf{x}) = \mathsf{x} \circledast \mathbf{W}_\rho + \boldsymbol{b}_\rho$$

Definition 3 (Modulator)**.**

$$\mu : \mathbb{R}^{H\times W\times C} \times \mathbb{R}^{H\times W\times C} \to (0,1)^{H\times W}$$
$$\mu(\mathsf{a},\, \mathsf{b}) = \sigma\left(\frac{\langle \boldsymbol{a}_{h,w},\, \boldsymbol{b}_{h,w}\rangle}{\|\boldsymbol{a}_{h,w}\|_2 \|\boldsymbol{b}_{h,w}\|_2}\right)_{(h,w)\in\{1,\ldots,H\}\times\{1,\ldots,W\}}$$

An algorithm for training is given in pseudocode as follows.

Algorithm 1 Train(x_t)

$\mathcal{M}_t \leftarrow \mathcal{M}_t \cup \{\mathsf{x}_t\}$
$\mathsf{x}_s \leftarrow \mathrm{rand}(\mathcal{M}_t)$
$\hat{\mathsf{x}}_s \leftarrow \rho(\mathsf{x}_s)$
$\mathsf{m} \leftarrow \mu(\hat{\mathsf{x}}_s,\, \mathsf{x}_t)$
return $\mathsf{m} \odot \hat{\mathsf{x}}_s + (1 - \mathsf{m}) \odot \mathsf{x}_t$

An algorithm for inference is given in pseudocode as follows.

Algorithm 2 Infer(x_t)

$\mathsf{x}_s \leftarrow \mathsf{x}_t$
$\hat{\mathsf{x}}_s \leftarrow \rho(\mathsf{x}_s)$
$\mathsf{m} \leftarrow \mu(\hat{\mathsf{x}}_s,\, \mathsf{x}_t)$
return $\mathsf{m} \odot \hat{\mathsf{x}}_s + (1 - \mathsf{m}) \odot \mathsf{x}_t$

The memory buffer maintains some number of feature maps for recollection. In practice, the memory buffer can be implemented as a double-ended queue of finite maximum length, for efficiency. Sampling from the memory buffer introduces inductive bias from any previously encountered examples, which may encode useful information. The randomness of this sampling can have a regularising effect on network activations, which could promote generalisation or mitigate overfitting.

The recollector is a learnable convolution for a recalled feature map. The modulator can be seen as a non-learnable factor which modulates between a current feature map and a recollected feature map. In particular, the modulator computes a cosine similarity between a current feature map and a recollected feature map, at each spatial position. This similarity is mapped to a soft attention-like score by a sigmoidal function. In practice, we used the sigmoid function simply for its smooth properties. Finally, the somnial unit returns a modulated linear combination between a current feature map and a recollected feature map. In practice, this interpolation between present and recalled information can then be passed to other (e.g. linear) layers in a neural network.

4 Methodology

4.1 Dataset

4.1.1 Description

We elected to construct a dataset based on the recyclability requirements stipulated in the NEA recycling guidelines [1]. Our current dataset comprises a total of 3118 images, equally split between 1559 recyclable (class 1) and 1559 non-recyclable (class 0) items. This class balance was used for each model to learn about each class in equal measure. Each image was hierarchically added according to the NEA guidelines. Firstly, we added images by material type: paper, plastic, glass, metal, and others. Secondly, we added images by their specific "types" as defined by the guidelines. Our dataset was compiled from the following sources.

Table 1: Dataset sources.

Dataset	Classes
TrashNet	6
RealWaste	9
Other sources	–

For each NEA-defined waste type, we fixed a minimum sample count to ensure their representation in the dataset.

4.1.2 Exploratory analysis

To inform modelling, we analysed the dataset across some of its features.

Table 2: Summary statistics for some features of the dataset.

Feature	Mean	Median
Height	469.91	500.00
Width	530.57	512.00

We noted that the smoothness of a surface, which could be telling of its state of contamination, could be useful information for classification.

4.1.3 Feature engineering

We implemented local binary pattern (LBP) feature engineering, augmenting a copy of the dataset as follows.

1. **Image to grayscale image.** Convert each image to a grayscale image via luminance-weighted averaging.
2. **Grayscale image to LBP image.** Compute the LBP using the "uniform" operator with $P = 8$ and radius $R = 1$.
3. **LBP image to normalised LBP image.** Divide the LBP image by its maximum value to obtain a normalised LBP image.
4. **Image and LBP to augmented image.** Concatenate the original image and the normalised LBP image channel-wise to obtain a 4-channel augmented image.

Each dataset was split into training, validation, and test sets in a ratio of $8 : 1 : 1$ in a class-stratified manner.

4.2 Models

We explored various architectures, including convolutional neural network (CNN), vision transformer (ViT), and hybrid architectures. These architectures are defined more precisely in the GitHub repository.

4.2.1 Training

On each training set, we trained various models using binary cross-entropy (CE) loss on the following settings, where settings not mentioned are assumed to use library defaults. The libraries we used are listed in the GitHub repository. For reproducibility, we seeded each function we used to the greatest extent practicable to our knowledge.

Call the maximum number of epochs without an improvement in validation loss (before early stopping), patience.

Table 3: Training hyperparameters for models.

Seed	37
Batch size	32
Optimiser	AdamW
Learning rate	0.001
Number of epochs	50
Patience	$\{10, 50\}$

Table 4: Training hyperparameters for models equipped with a somnial unit.

Memory buffer size	$\{5, 10, 100\}$

We also considered using additional algorithms, such as genetic and simulated annealing algorithms, to look for more optimal hyperparameters. However, we did not do so owing to hardware limitations and time constraints.

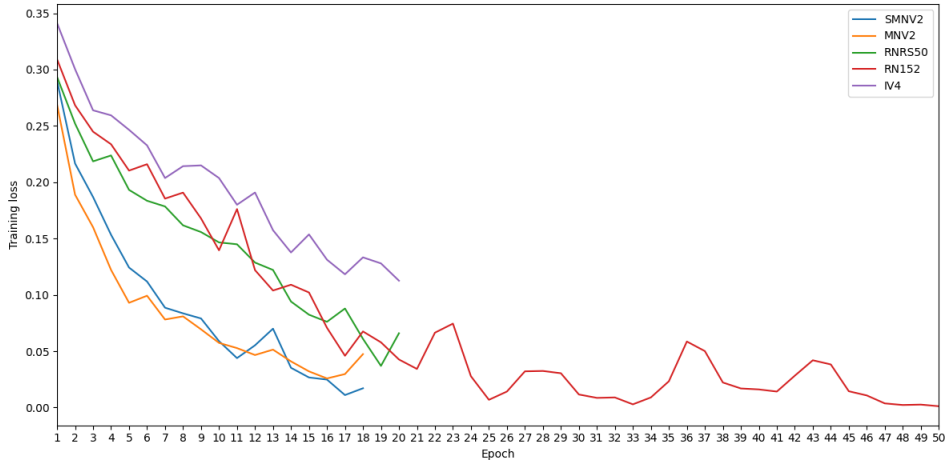

Figure 1: Training losses for models.

4.2.2 Evaluation

On each test set, we evaluated the trained models using the macro F1 metric. We used this metric to measure the performance of each model in a balanced manner across both classes. Our top-performing model was evaluated with and without a somnial unit, to investigate the effects of somnial unit addition. For brevity, the identifiers (IDs) of models equipped with a somnial unit are prefixed with "S", and only a selection of our models are shown.

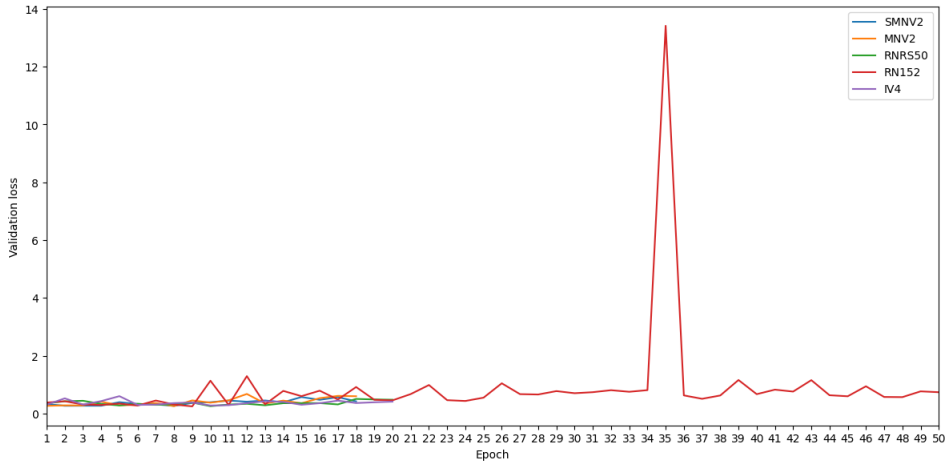

Figure 2: Validation losses for models.

Table 5: Evaluation results for models.

ID	Family	Macro F1
SMNV2	CNN	0.945387
MNV2	CNN	0.935832
RNRS50	CNN	0.922874
RN152	CNN	0.926111
IV4	CNN	0.919354
ViTB32	CNN	0.869007
ViTB16	CNN	0.836259

We also tried many more models: a list of the other models tried can be found in a later section.

5 Discussion

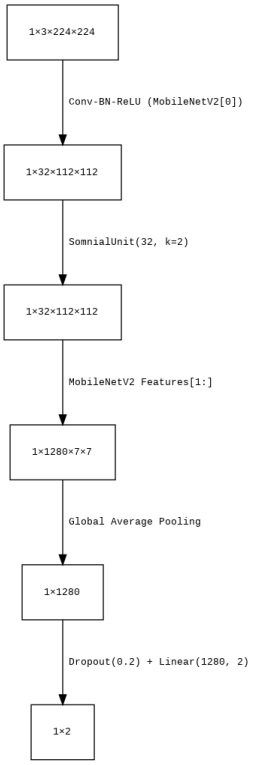

Figure 3: Architecture of SMNV2.

We consider SMNV2 our best model as it achieved the highest scores on the metric and took the least time for inference.

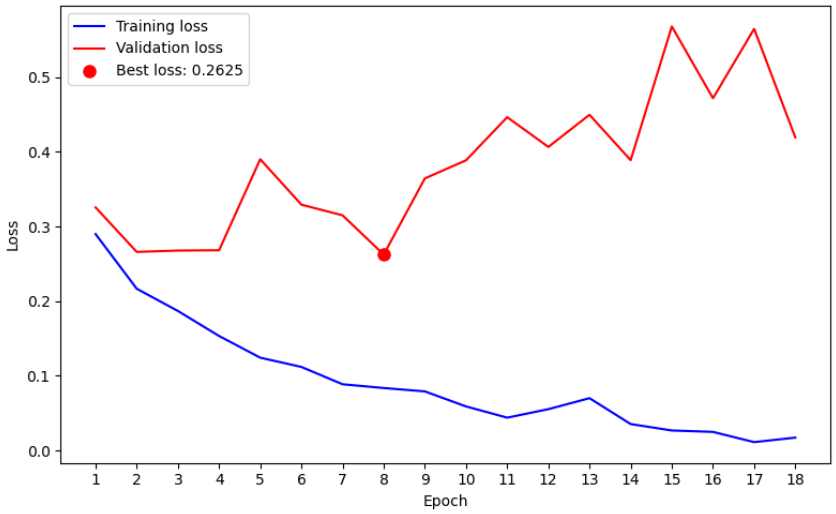

Figure 4: Training and validation losses for SMNV2.

Notably, the addition of a somnial unit to MNV2 slightly improved its performance. Across the board, model families can also be ranked by their macro F1 scores. The CNN models significantly outperformed the ViT models. We conclude that the inductive bias introduced by convolution was useful for the task. Overall, we are fairly certain that more investigation could yield a better understanding of the results we obtained. Due to hardware limitations and time constraints, we were only able to run ablative studies for the somnial unit on some of our simplest models, including our top-performing model.

We used a gradient-weighted class activation mapping (Grad-CAM) to try to generate a visual explanation for the predictions made by MNV2 on its test set.

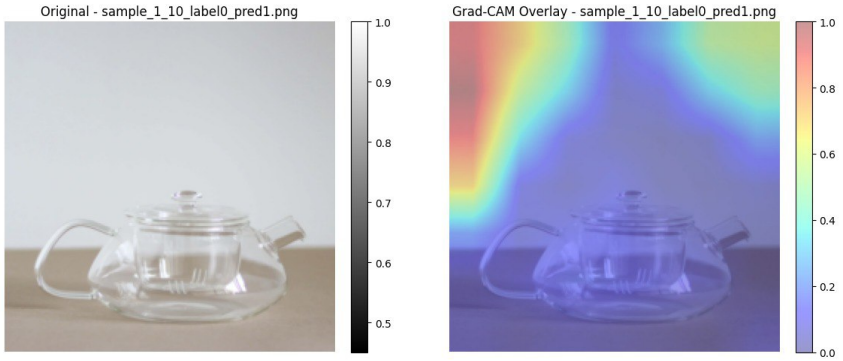

Figure 5: Grad-CAM visualisation of a false positive predicted by MNV2.

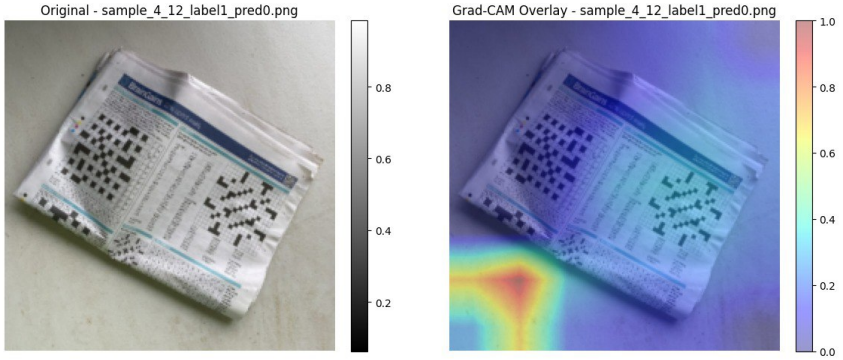

Figure 6: Grad-CAM visualisation of a false negative predicted by MNV2.

The model does not appear to be paying much attention to the objects' surfaces. We suspect that this result can be attributed to the small size and "noisiness" of our constructed dataset. Most of our recyclables derive from TrashNet, which features relatively "cleaner" background surfaces, while most of our non-recyclables derive from RealWaste, which features relatively "dirtier" background surfaces.

6 Failed Attempts

In this section, we list the various approaches and models we tried, which did not nearly produce the same level of performance shown in the previous section.

- Use of class-weighted cross-entropy loss.
- Use of focal loss.
- Use of mixup.
- Feature fusion between best models.
- Feature concatenation between best models.
- Addition of squeeze-excitation (SE) block to MNV2 and RN152.
- Self-distillation of MNV2.
- Trial of ConvNeXt, EVA02, DINOv2, CoaT, MobileViT, and NeXt-ViT.
- Curriculum learning using Laplacian variance as measure of difficulty.

7 Future Work

The somnial unit was developed iteratively in both a breadth-wise and depth-wise fashion. On one iteration of the unit, we replaced the convolutional layer of the recollector with a Fourier neural operator (FNO). However, this increased memory usage, which eventually made training infeasible on our hardware. On another iteration of the unit, we defined the modulator as a convolutional layer which learns a concatenation of a current feature map with a recollected feature map. However, we eventually scrapped this idea to reduce the computational complexity of the unit. Briefly, we also explored using Kolmogorov–Arnold layers, cyclic topologies, Fourier feature mapping, and an adversarial approach to boundary refinement.

The somnial unit admits several modifications and extensions as follows.

- For a classification task, one could define a memory buffer for each class, so that for each class c, a memory buffer for c stores samples which belong only to class c.
- One could use a sampling function other than a uniform pseudorandom selection function.
- One could noise the generator by an amplitude to introduce regularisation.
- One could use a moving average (e.g. exponential moving average) to modulate between current and recollected feature maps.

To improve our best model's performance on the task, one may also explore temperature scaling, spatial dropout, and various other adversarial methods, in the future. With access to more data and compute, one may also assess the generalisability of our best models across more diverse settings.

8 Conclusion

In this paper, we introduce a novel architectural unit we call a *somnial unit*, and demonstrate its use on a waste recyclability task unique to Singapore.

References

[1] National Environment Agency. *List of Items That Are Recyclable and Not*. `https://www.nea.gov.sg/docs/default-source/our-services/waste-management/list-of-items-that-are-recyclable-and-not.pdf`. Accessed: April 22, 2025. National Environment Agency, Singapore.

[2] Hongyu Zheng and Yong Gu. "EnCNN-UPMWS: Waste Classification by a CNN Ensemble Using the UPM Weighting Strategy". In: *Electronics* 10.4 (2021), p. 427. ISSN: 2079-9292. DOI: 10.3390/electronics10040427. URL: `https://www.mdpi.com/2079-9292/10/4/427`.