Declaration of Academic Integrity

Module Code & Module Name	ET0732 MACHINE LEARNING & AI (MLAI)
Assessment (e.g. Mini Project)	Mini Project

Declaration of Academic Integrity

Academic Integrity is a central tenet of Singapore Polytechnic. The polytechnic rules state that "*Cheating in examinations and other assessed work is a very serious offence. This includes copying and using plagiarised material. Any student who cheats, attempts to cheat or breaches any rules for examinations and tests will face disciplinary action. The student is liable to be expelled.*"

Check only one of the two options below:

☐ I/We affirm that the work I/we submit is my/our own, produced without help from any AI tool(s).

☐ I/We affirm that the work I/we submit has been produced with the use of AI tool(s) which I/we have acknowledged fully.

By signing this form, I/we declare that the above affirmation made is true, and that I/we have read and understood the rules stated in Students Handbook on "Plagiarism" and "Breach of Examination/Assessment Rules".

#	Student ID	Name	Signature	Date
1	P2319104	Bryan Ong Jia Le		22 Jan 26
2	P23464238	Lee Ei Zin		30 Jan 26
3				
4				
5				

Important:

Students will not incur penalties for the utilisation of AI tools in their assignments, provided that due acknowledgment is given, and wholesale replication of the AI tools' output is abstained from. Instances where students incorporate AI tools in their assignments and replicate the generated output wholesale, or fail to comprehensively disclose the use of AI tools, will be deemed as instances of academic dishonesty and will be subject to appropriate disciplinary measures.

Acknowledgement

To acknowledge the use of an AI tool, use the template below. Fill in the name of the AI tool, your name, your input prompts, date, and output generated by the AI tool. Extend the template if needed.

we just paste some random prompts in here later on (will delete this line when we done)

<table>
<tr><td>Name of AI tool</td><td>Perplexity</td></tr>
<tr><td>Name of student</td><td>Bryan</td></tr>
<tr><td>Input prompt</td><td>whatre the benefits of only data augmentation of training, not testing or val over data augmentation of all 3? doesnt accuracy not matter as long as the model is at its peak performance?</td></tr>
<tr><td>Date generated</td><td>29 Jan 2026</td></tr>
<tr><td>Output generated</td><td>
<p>Augmenting all 3 inflates metrics artificially—"peak performance" on fake data doesn't predict real-world success.</p>
<h2>Why Train-Only Wins</h2>
<table>
<tr><th>Strategy</th><th>Train Acc</th><th>Val Acc</th><th>Real-World Acc</th></tr>
<tr><td>Train aug only ✓</td><td>95%</td><td>85% (honest gap)</td><td>~85% (matches val)</td></tr>
<tr><td>All 3 augmented</td><td>95%</td><td>92% (fake high)</td><td>75% (drops off cliff)</td></tr>
</table>
<h2>Critical Reasons</h2>
<ol>
<li>Val/Test = Reality Proxy: Clean originals mimic production photos. Augmented val = "your model crushes rotated test photos"—useless for straight-on factory cams.</li>
<li>Overfitting Detection: Train-val gap shows generalization. No gap? Model memorized augmented tricks, not lids.</li>
</ol>
</td></tr>
</table>

	3. No Data Leak: Augmentation simulates variety your model *must* handle—but val/test stay fixed for unbiased scoring. Counterpoint: Accuracy matters for deployed performance, not training logs. Train-only gives trustworthy val/test numbers matching your cup noodle factory accuracy. Industry standard.
Impact on submission	Allowed me to understand the model better.

Name of AI tool	Perplexity
Name of student	Bryan
Input prompt	whhat form of data visualization could I do for my CNN model? list from most impt to least (code was attached)
Date generated	31 Jan 2026
Output generated	Here are the plots I'd prioritize for your project, from most important (almost mandatory) to more "nice to have". --- **Core training curves (highest priority)** 1. Training vs validation loss over epochs - Shows if the model is overfitting (val loss going up while train loss keeps going down). - X-axis: epochs, Y-axis: loss, 2 lines: `loss` and `val_loss` from `history.history`. 2. Training vs validation accuracy over epochs - Shows how performance improves and where it plateaus. - X-axis: epochs, Y-axis: accuracy, 2 lines: `accuracy` and `val_accuracy`. 3. Learning rate vs epochs (if you log LR) - Since you use `ReduceLROnPlateau`, a line plot of LR vs epoch helps you show how LR changed when validation stagnated. --- **Evaluation-focused plots (on test set)** 4. Confusion matrix for test set - 2D heatmap with true class on Y-axis and predicted class on X-axis, shows where the model confuses classes.

   - Very good for explaining model behavior in a report.
5. Per-class accuracy bar chart
   - Bar chart where each bar is a class and height is accuracy (or recall) for that class.
   - Shows if some classes (e.g. minority ones) are harder to classify.
6. Support (number of samples) per class bar chart
   - Bar chart of number of images per class in your train/val/test sets.
   - Good to visually justify why you used `class_weight`.

Data and augmentation visuals

7. Sample images per class (grid)
   - A 3×3 or 4×4 grid showing random images, labelled with their class.
   - Helps the marker see what your data looks like.
8. Augmented vs original images
   - Show 1 original image and several augmented versions (rotation, shift, zoom, flip).
   - Visually explains what `ImageDataGenerator` is doing.
9. Class distribution pie chart or bar chart (overall dataset)
   - This can be one plot for total dataset, or three separate plots for train/val/test distribution.

More advanced / optional plots

10. Precision, recall, F1 per class (bar chart)
    - Use metrics from `classification_report` and plot as grouped bars (precision/recall/F1) by class.
11. ROC curves (if turned into one-vs-rest for multi-class)
    - Multi-line plot of ROC curves per class, with AUC in the legend.
12. Prediction confidence histogram

	- Histogram of max softmax probability per test sample, to show if the model is usually confident or uncertain. 13. Misclassified examples grid - Show some images the model got wrong, with “true: X, pred: Y, conf: Z” below each. 14. Training time per epoch line plot - X-axis: epoch, Y-axis: seconds; shows if training gets slower or stays stable. --- **What I’d definitely include for coursework** If you need to be efficient but still look impressive, I’d do at least: - Training vs validation loss curve. - Training vs validation accuracy curve. - Confusion matrix on test set. - Class distribution bar chart (train/val/test). - One figure of original vs augmented images, to show data augmentation. If you want, I can next give you exact Matplotlib code snippets using your `history`, `train_gen`, `val_gen`, and `test_gen` to generate each of these.
Impact on submission	Allowed me to take inspiration from some of these data visualization methhhods to add them to our ipynb.

Name of AI tool	ChatGPT
Name of student	*Ei Zin*
Input prompt	what is the purpose of data augmentation in terms of rotation, width & height shift, zoom and horizontal flip?
Date generated	30/1/2026
Output generated	To **make your model generalize better** by teaching it that these variations don’t change the class. Quick breakdown: - **Rotation**: Handles images taken at slightly different angles. Model learns orientation isn’t important. - **Width / Height shift**: Simulates objects appearing off-center. Prevents overfitting to centered subjects.

	● **Zoom**: Teaches the model to recognize objects at different scales/distances. ● **Horizontal flip**: Makes the model invariant to left–right orientation (use only if flipping doesn't change the meaning). Overall: **more effective training data without collecting more images**, less overfitting, better real-world performance. If you want, I can tell you when *not* to use each one (very important for some datasets).
Impact on submission	It helped me decide whether or not we should include data augmentation to our model with a small dataset