Criteria	weight	Failed (0)	Passed (50)	Good (70)	Excellent (100)
Data preparation and visualisation	5%	Not attempted or incomplete	• Dataset split correctly to form a validation set • A function taking in appropriate arguments for display 20 random images from each set; function was there but needed some modification to run.	• Dataset split correctly to form a validation set • A function taking in appropriate arguments for display 20 random images from each set was found in the code and ran well.	• Dataset split correctly to form a validation set • A function taking in appropriate arguments for display 20 random images from each set was found in the code and ran well; function was well documented; images displayed were labelled with class names.
MLP (a) **Hyperparameters**	15%	Not attempted or incomplete	Markdown cell(s) were given with detailed explanation about the hyperparameter tuning process. Experiments with two possible settings for **one** of the following: • **connection weight initialisation**; • **learning rate scheduling** (including a learning rate calculation function and call back); • **dropout rate**. It was not certain from the explanation that the procedure was carried out.	Markdown cell(s) were given with detailed explanation about the hyperparameter tuning process. Experiments with two possible settings for **two** of the following: • **connection weight initialisation**; • **learning rate scheduling** (including a learning rate calculation function and call back); • **dropout rate**. Some minimal explanation was supplied.	Markdown cell(s) were given with detailed explanation about the hyperparameter tuning process. Experiments with two possible settings for **all** of the following: • **connection weight initialisation**; • **learning rate scheduling** (including a learning rate calculation function and call back); • **dropout rate**. **Early stopping** (through callback) was incorporated. The validation set was appropriately used. Procedure was well explained and easy to follow.
MLP (b) **Architecture**	6%	Not attempted or incomplete	• Basic code was there but it needed some modification in order to run. • The MLP had 2-3 hidden layers; each layer had an appropriate number of neurons and activation function. • A brief explanation on the network architecture was included.	• Basic code was there and the code ran without problems. • The MLP had 2-3 hidden layers; each layer had an appropriate number of neurons and activation function. • A brief explanation on the network architecture was included.	• Basic code was there and the code ran without problems. • The MLP had 2-3 hidden layers; each layer had an appropriate number of neurons and activation function • Other settings were **consistent** with the investigation on the hyperparameters (part (a) above). • Information about the network was displayed. A brief explanation was included.
MLP (c) **Training and Testing**	12%	Not attempted or incomplete	• Network compiled okay. • Training and prediction steps looked okay but code did not run somehow. • There was a lack of explanation in the code or markdown cells. • Code was messy.	• Network compiled okay. • Network could **(1)** be trained in full from scratch and the trained model was saved successfully; and **(2)** load a pretrained model and be trained for 1 epoch. • An optimizer and an appropriate loss function were used and a brief explanation was given. • Training and prediction steps ran ok.	• Network compiled okay. • Network could **(1)** be trained in full from scratch and the trained model was saved successfully; and **(2)** load a pretrained model and be trained for 1 epoch. • An optimizer and an appropriate loss function were used and a brief explanation was given. • Validation set was appropriately used. • Training and prediction steps ran successfully to completion. • Code was well explained.
MLP (d) **Classification results**	8%	Not attempted or incomplete	• Classification accuracy and F1 score on the test set were provided with some explanation. • Code was messy.	• Classification accuracy and F1 score on the test set were provided with some explanation. • A confusion matrix for the test set was shown (graphically). • One or two classification results were shown.	• Classification accuracy and F1 score on the test set were provided with good explanation. • A confusion matrix for the test set was shown (graphically) with clear class labels. • A few correctly classified images and a few failure cases were shown with explanation.

CNN (a) **Hyperparameters**	20%	Not attempted or incomplete	Markdown cell(s) were given with detailed explanation about the hyperparameter tuning process. Experiments with two possible settings for **one** of the following: • **kernel size**, • **number of kernels**, • **activation function** It is not evident that the experiments were indeed carried out.	Markdown cell(s) were given with detailed explanation about the hyperparameter tuning process. Experiments with two possible settings for **two** of the following: • **kernel size**, • **number of kernels**, • **activation function** Some minimal explanation was supplied.	Markdown cell(s) were given with detailed explanation about the hyperparameter tuning process. Experiments with two possible settings for **all** of the following: • **kernel size**, • **number of kernels**, • **activation function** Early stopping (through callback) was incorporated. The validation set was appropriately used. Procedure was well explained and easy to follow.
CNN (b) **Architecture**	6%	Not attempted or incomplete	•Basic code was there and the code seemed to work. • The CNN had 2-3 convolutional layers and some pooling layers. .	• Basic code was there and the code ran without problems. • The CNN had 2-3 convolutional layers; each layer had an appropriate number of neurons and activation function. • The CNN had some pooling layers and batch normalisation. • A brief explanation on the network architecture was included.	• Basic code was there and the code ran without problems. • The CNN had 2-3 convolutional layers; each layer had an appropriate number of neurons and activation function. • The CNN had some pooling layers and batch normalisation. • Other settings were **consistent** to the investigation on the hyperparameters (part (a) above). • Information about the network was displayed. A brief explanation was included.
CNN (c) **Training and testing**	12%	Not attempted or incomplete	• Network compiled okay. • Training and prediction steps looked okay but code did not run somehow.	• Network compiled okay. • Network could **(1)** be trained in full from scratch and the trained model was saved successfully; and **(2)** load a pretrained model and be trained for 1 epoch. • An optimizer and an appropriate loss function were used and a brief explanation was given. • Training and prediction steps ran ok.	• Network compiled okay. • Network could **(1)** be trained in full from scratch and the trained model was saved successfully; and **(2)** load a pretrained model and be trained for 1 epoch. • An optimizer and an appropriate loss function were used and a brief explanation was given. • Validation set was appropriately used. • Training and prediction steps ran successfully to completion. • Code was well explained.
CNN (d) **Classification results**	8%	Not attempted or incomplete	• Classification accuracy and F1 score on the test set were provided with some explanation. • Code was messy.	• Classification accuracy and F1 score on the test set were provided with some explanation. • A confusion matrix for the test set was shown (graphically). • One or two classification results were shown.	• Classification accuracy and F1 score on the test set were provided with good explanation. • A confusion matrix for the test set was shown (graphically). • A few correctly classified images and a few failure cases were shown with explanation.
MLP & CNN **Comparison & Summary**	8%	Not attempted or incomplete	Brief comparison on the classification accuracy and confusion matrices.	(markdown cells and (optional) code for illustration) Brief comparison on the classification accuracy between MLP and CNN in terms of **three** of the following: classification accuracies, confusion matrices, network architectures (model's complexity), examples/classes where one model worked but the other failed, classes where both worked well or failed, training time.	(markdown cells and (optional) code for illustration) Comprehensive comparison on the classification accuracy between MLP and CNN in terms of **at least four** of the following: classification accuracies, confusion matrices, network architectures (model's complexity), examples/classes where one model worked but the other failed, classes where both worked well or failed, training time.
	100.00%				

Penalty on top of late submissions:

10.00% If intermediate models were submitted, making the zip file too large to download
10.00% If the zip file included the dataset files
10.00% If the model was much more complicated than the network architecture specified in the labsheet.