

Criteria	weight	Failed (0)	Passed (50)	Good (70)	Excellent (100)
Data preparati on and visualisati on	7%	Not attempted or incomplete	<ul style="list-style-type: none"> Dataset split correctly to form a validation set 20 random images from each set were displayed but the code was not completely correct. 	<ul style="list-style-type: none"> Dataset was split correctly to form a validation set A function taking in appropriate arguments for <i>correctly</i> display 20 random images from each set was found in the code and ran well. 	<ul style="list-style-type: none"> Dataset was split correctly to form a validation set A function taking in appropriate arguments for <i>correctly</i> displaying 20 random images from each set was found in the code and ran well; function was well documented; images were displayed meaningfully.
MLP (a) Hyperpar ameters	15%	Not attempted or incomplete	<p>Markdown cell(s) were given with detailed explanation about the hyperparameter tuning process. Experiments with two possible settings for two of the three hyperparameters.</p> <p>It was not certain from the explanation that the hyperparameter procedure was carried out.</p>	<p>Markdown cell(s) were given with detailed explanation about the hyperparameter tuning process. Experiments with two possible settings for two of the three hyperparameters.</p> <p>Some minimal explanation was supplied.</p>	<p>Markdown cell(s) were given with detailed explanation about the hyperparameter tuning process. Experiments with two possible settings for all of the three hyperparameters.</p> <p>Early stopping (through callback) was incorporated.</p> <p>The validation set was appropriately used. Procedure was well explained and easy to follow.</p>
MLP (b) Architect ure	6%	Not attempted or incomplete	<ul style="list-style-type: none"> 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. 	<ul style="list-style-type: none"> Basic code was there and the code ran without problems. The MLP had the correct architecture as specified in the labsheet; each layer had an appropriate number of neurons and activation function. A brief explanation on the network architecture was included. 	<ul style="list-style-type: none"> Basic code was there and the code ran without problems. The MLP had the correct architecture as specified in the labsheet; 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	9%	Not attempted or incomplete	<ul style="list-style-type: none"> 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 markdwn cells. Code was messy. 	<ul style="list-style-type: none"> Network compiled okay. The code could (1) train the model from scratch and can correctly save the trained model; and (2) load a pretrained model and train it for 1 epoch, with the correct learning rate. An optimizer and an appropriate loss function were used and a brief explanation was given. Training and prediction steps ran ok. 	<ul style="list-style-type: none"> Network compiled okay. The code could (1) train the model from scratch using the training and validation sets and can correctly save the trained model; and (2) load a pretrained model and train it for 1 epoch, with the correct learning rate. An optimizer and an appropriate loss function were used and a brief explanation was given. Training and prediction steps ran successfully to completion. Code was well explained.
MLP (d) Classificat ion results	8%	Not attempted or incomplete	<ul style="list-style-type: none"> Some of the required classification measures on the test set were reported. Code was messy. 	<ul style="list-style-type: none"> All the required classification measures on the test set were reported. Some illustrations were shown. 	<ul style="list-style-type: none"> All the required classification measures on the test set were reported. Results were well explained. Illustrations were meaningful and clear.
CNN (a) Hyperpar ameters	15%	Not attempted or incomplete	<ul style="list-style-type: none"> Markdown cell(s) were given with some explanation about the hyperparameter tuning process. Experiments with two possible settings for all of the three hyperparameters were described. <p>It is not evident that the experiments were indeed carried out.</p>	<ul style="list-style-type: none"> Markdown cell(s) were given with detailed explanation about the hyperparameter tuning process. Experiments with two possible settings for all of the three hyperparameters were described clearly. <p>The validation set was appropriately used. Procedure was explained.</p>	<ul style="list-style-type: none"> Markdown cell(s) were given with detailed explanation about the hyperparameter tuning process. Experiments with two possible settings for all of the three hyperparameters were described clearly. Early stopping (through callback) was incorporated. The validation set was appropriately used. Procedure was well explained and easy to follow.
CNN (b) Architect ure	6%	Not attempted or incomplete	<ul style="list-style-type: none"> Basic code was there and the code seemed to work. The CNN had 2-3 convolutional layers and some pooling layers. 	<ul style="list-style-type: none"> 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. 	<ul style="list-style-type: none"> 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	9%	Not attempted or incomplete	<ul style="list-style-type: none"> Network compiled okay. Training and prediction steps looked okay but code did not run somehow. 	<ul style="list-style-type: none"> 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. 	<ul style="list-style-type: none"> Network compiled okay. The code could (1) train the model from scratch using the training and validation sets and save the trained model correctly; and (2) load a pretrained model and be trained for 1 epoch, with the correct learning rate. An optimizer and an appropriate loss function were used and a brief explanation was given. Training and prediction steps ran successfully to completion. Code was well explained.
CNN (d) Classificat ion results	8%	Not attempted or incomplete	<ul style="list-style-type: none"> Some of the required classification measures on the test set were reported. Code was messy. 	<ul style="list-style-type: none"> All of the required classification measures on the test set were reported. Prediction results on the test set were shown. 	<ul style="list-style-type: none"> All of the required classification measures on the test set were reported. Prediction results on the test set were well explained. Illustrations were meaningful and clear.
MLP & CNN Comparis on & Summary	17%	Not attempted or incomplete	Brief comparison on the classification accuracy and confusion matrices.	(markdown cells and code for illustration) Brief comparison on the classification accuracy between MLP and CNN in terms of some 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, clear explanation.	(markdown cells and code for illustration) Comprehensive comparison on the classification accuracy between MLP and CNN in terms of all 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, clear explanation (e.g., why each model worked or did not work well).

100.00%

NOTE: Do not simply implement minimal coding based on the last column shown above and expect to get 100%. In the marking process, we also take into account the overall layout, neatness, etc. of your code. For example, you can lose marks (dropping from *Excellent* down to *Good* or *Passed*) if your code is messy (e.g., many small code cells seem to be for debugging purpose), the *print* statements are not meaningful, the variable names are not meaningful, diagrams are unclear or not meaningful. We also look for appropriate use of for loops and helper functions where appropriate (rather than simply duplicating the code for doing similar things over several code cells).