

Criteria	weight	Failed (0)	Passed (50)	Good (70)	Excellent (100)
<b>MLP (a) Hyperparameters</b>	15%	Not attempted or incomplete	Markdown cell(s) were given about the hyperparameter tuning process. Experiments with <b>one</b> of the following: • <b>connection weight initialisation;</b> • <b>learning rate scheduling;</b> • <b>early stopping</b> It is not evident that the experiments were indeed carried out.	Markdown cell(s) were given about the hyperparameter tuning process. Experiments with <b>two</b> possible settings for two of the following: • <b>connection weight initialisation;</b> • <b>learning rate scheduling;</b> • <b>early stopping</b> using either grid search or customized code (shown in Markdown cells). An appropriate validation set was used.	Markdown cell(s) were given with detailed explanation about the hyperparameter tuning process. Experiments with two possible settings for <b>all</b> of the following: • <b>connection weight initialisation;</b> • <b>learning rate scheduling;</b> • <b>early stopping</b> using either grid search or customized code (shown in Markdown cells). An appropriate validation set was used. Explanation about the experiments was supplied.
<b>MLP (b) Architecture</b>	4%	Not attempted or incomplete	Basic code was there and the code ran without problems. The MLP has two hidden layers; each layer has an appropriate number of neurons and activation function.	Basic code was there and the code ran without problems. The MLP has two hidden layers; each layer has 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 has two hidden layers; each layer has an appropriate number of neurons and activation function. Other settings were consistent to the investigation on the hyperparameters (part (a) above). A brief explanation on the network architecture was included.
<b>MLP (c) Training and Testing</b>	8%	Not attempted or incomplete	Network compilation ran okay. Training and prediction steps looked okay but code did not run somehow.	Network compilation ran okay. Training and prediction steps ran successfully to completion.	Network compilation ran okay. Appropriate optimizer and loss function were used and a brief explanation was given. Training and prediction steps ran successfully to completion.
<b>MLP (d) Classification results</b>	10%	Not attempted or incomplete	Classification accuracy on the test set was provided with some explanation.	Classification accuracy on the test set was provided with some explanation. A confusion matrix for the test set was shown (graphically). One or two classification results were shown.	Classification accuracy on the test set was 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.
<b>CNN (a) Hyperparameters</b>	20%	Not attempted or incomplete	Markdown cell(s) were given with detailed explanation about the hyperparameter tuning process. Experiments with two possible settings for <b>one</b> of the following: • <b>kernel size,</b> • <b>number of kernels,</b> • <b>dropout rate,</b> • <b>activation function</b> 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 <b>two</b> of the following: • <b>kernel size,</b> • <b>number of kernels,</b> • <b>dropout rate,</b> • <b>activation function</b> using either grid search or customized code (shown in Markdown cells). An appropriate validation set was used.	Markdown cell(s) were given with detailed explanation about the hyperparameter tuning process. Experiments with two possible settings for <b>all</b> of the following: • <b>kernel size,</b> • <b>number of kernels,</b> • <b>dropout rate,</b> • <b>activation function</b> using either grid search or customized code (shown in Markdown cells). An appropriate validation set was used. Explanation about the experiments was supplied.
<b>CNN (b) Architecture</b>	5%	Not attempted or incomplete	Basic code was there and the code ran without problems. The CNN has two convolutional layers.	Basic code was there and the code ran without problems. The CNN has two convolutional layers, pooling layers, and other appropriate layers. A brief explanation on the network architecture was included.	Basic code was there and the code ran without problems. The CNN has two convolutional layers, pooling layers, and other appropriate layers. The network has the optimal parameters worked out in part (a). A brief explanation on the network architecture was included.
<b>CNN (c) Model saving and restoration</b>	9%	Not attempted or incomplete	User was given the option to load, train, and save model. The model loading and model saving operations did not work 100%.	User was given the option to load, train, and save model. One of the following operation worked: <b>Model loading, Model saving.</b>	User was given the option to load, train, and save model. Both of the following operations worked: <b>Model loading:</b> Pre-trained model and all the associated files could be loaded successfully. <b>Model saving:</b> trained model (part (c) below) could be saved to disk.
<b>CNN (d) Training and testing</b>	9%	Not attempted or incomplete	Network compilation ran okay. Training and prediction steps looked okay but code did not run somehow.	Network compilation ran okay. Training and prediction steps ran successfully to completion.	Network compilation ran okay. Appropriate optimizer and loss function were used and a brief explanation was given. Training and prediction steps ran successfully to completion.
<b>CNN (e) Classification results</b>	12%	Not attempted or incomplete	Classification accuracy on the test set was provided with some explanation (if training and testing ran okay before).	Classification accuracy on the test set was provided with some explanation. A confusion matrix for the test set was shown (graphically). One or two classification results were shown.	Classification accuracy on the test set was 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.
<b>MLP &amp; CNN Comparison</b>	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.
Total	100%				

**Note:** Your Python code does not need to follow the same order of the different parts of the marking rubric given above. e.g., your code might have MLP (a) appears before MLP (b).