

Team ID: PNT2022TMID24591
A Novel Method For Handwritten Digit Recognition System

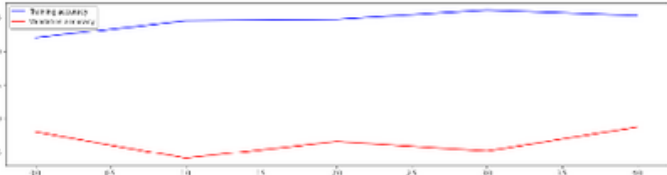
Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Data Collection	USN-1	As a user, I can collect the dataset from various resources with different handwritings.	10	Low
Sprint-1	Data Preprocessing	USN-2	As a user, I can load the dataset, handling the missing data, scaling and split data into train and test.	10	Medium
Sprint-2	Model Building	USN-3	As a user, I will get an application with ML model which provides high accuracy of recognized handwritten digit.	5	High
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layers to it.	5	High
Sprint-2	Compiling the model	USN-5	With both the training data defined and model defined, it's time to configure the learning process.	2	Medium
Sprint-2	Train & test the model	USN-6	As a user, let us train our model with our image dataset.	6	Medium
Sprint-2	Save the model	USN-7	As a user, the model is saved & integrated with an android application or web application in order to predict something.	2	Low
Sprint-3	Building UI Application	USN-8	As a user, I will upload the handwritten digit image to the application by clicking a upload button.	5	High
Sprint-3		USN-9	As a user, I can know the details of the fundamental usage of the application.	5	Low
Sprint-3		USN-10	As a user, I can see the predicted / recognized digits in the application.	5	Medium
Sprint-4	Train the model on IBM	USN-11	As a user, I train the model on IBM and integrate flask/Django with scoring end point.	10	High
Sprint-4	Cloud Deployment	USN-12	As a user, I can access the web application and make the use of the product from anywhere.	10	High

Team ID: PNT2022TMID24591
A Novel Method For Handwritten Digit Recognition System

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot																								
1.	Model Summary	- Model: "sequential_1"	<pre>Model: "sequential" Layer (type) Output Shape Param # ----- conv2d (Conv2D) (None, 28, 28, 64) 640 conv2d_1 (Conv2D) (None, 24, 24, 32) 16464 flatten (Flatten) (None, 16432) 0 dense (Dense) (None, 10) 164330 ----- Total params: 203,434 Trainable params: 203,434 Non-trainable params: 0 ----- None</pre>																								
2.	Accuracy	Training Accuracy - 0.9805333614349365 Validation Accuracy - 0.9735000133514404	 <table border="1"><caption>Accuracy Data Points (Estimated from Graph)</caption><thead><tr><th>Epoch</th><th>Training Accuracy</th><th>Validation Accuracy</th></tr></thead><tbody><tr><td>0</td><td>0.975</td><td>0.975</td></tr><tr><td>5</td><td>0.978</td><td>0.972</td></tr><tr><td>10</td><td>0.980</td><td>0.970</td></tr><tr><td>15</td><td>0.982</td><td>0.972</td></tr><tr><td>20</td><td>0.983</td><td>0.973</td></tr><tr><td>25</td><td>0.984</td><td>0.974</td></tr><tr><td>30</td><td>0.985</td><td>0.975</td></tr></tbody></table>	Epoch	Training Accuracy	Validation Accuracy	0	0.975	0.975	5	0.978	0.972	10	0.980	0.970	15	0.982	0.972	20	0.983	0.973	25	0.984	0.974	30	0.985	0.975
Epoch	Training Accuracy	Validation Accuracy																									
0	0.975	0.975																									
5	0.978	0.972																									
10	0.980	0.970																									
15	0.982	0.972																									
20	0.983	0.973																									
25	0.984	0.974																									
30	0.985	0.975																									