Report on
Bangla Character Recognition Challenge

Submitted by
Rasman Mubtasim Swargo
Student ID: 1705051

1 Project Setup

1. CPU: AMD Ryzen 5 3600
2. RAM: 24 GB
3. Editor: Pycharm 2022.1.1

2 Preprocessing

1. Conversion to Grayscale
2. Color Ineversion
3. Resize

3 Result

The experiment was done using LeNet. Three learning rates, (0.01, 0.001, 0.0001), were examined for 15 epochs. Here are the results.

3.1 Table for Learning Rate = 0.01

Epoch	Train Loss	Training Accuracy	Training F1 Score	Validation Loss	Validation Accuracy	Validation F1 Score
1	78608.8278	47.75269817	0.494243792	6354.377945	76.03697024	0.759901184
2	72230.4879	52.56572829	0.550928326	4252.418916	84.38908927	0.842161187
3	73850.4879	52.47837236	0.548629555	4035.799598	85.16681695	0.849720206
4	82435.0604	53.29275509	0.564094128	3412.460508	87.84941389	0.877591421
5	90969.4441	53.71544509	0.573223061	3173.506879	88.96528404	0.889070866
6	89844.2564	55.11595796	0.588227391	3079.932064	89.31469793	0.892553368
7	87004.4207	57.77326908	0.606025735	2973.358789	90.0135257	0.899668134
8	88457.3366	57.89162228	0.612003845	2954.670665	89.74301172	0.897034463
9	91878.8519	59.00752388	0.626366517	2914.551552	90.28403968	0.902468156
10	87207.0411	62.34677488	0.649425513	2969.855316	90.1600541	0.901466073
11	90752.8315	59.98252881	0.638685834	2986.899498	90.45311091	0.904100957
12	76756.1489	65.4521374	0.676993844	3023.866806	90.53201082	0.904910666
13	98382.4485	60.49539268	0.652015469	3212.386606	90.374211	0.903400852
14	81862.8503	63.56130414	0.668929554	3279.96804	90.4418395	0.904067916
15	88341.4931	62.09034294	0.667084776	3330.549576	90.46438233	0.904420182

3.2 Table for Learning Rate = 0.001

Epoch	Train Loss	Training Accuracy	Training F1 Score	Validation Loss	Validation Accuracy	Validation F1 Score
1	194925.496	21.35148082	0.215510554	16782.94496	35.33588819	0.339060879
2	274103.872	28.51466734	0.312427005	13038.16682	51.43146979	0.509417659
3	273216.35	32.9416406	0.369453093	10286.12801	60.89945897	0.605082636
4	248856.082	35.95682926	0.408040647	8556.77718	67.63976555	0.675510228
5	220507.294	37.82230112	0.429757965	7548.601236	71.31424707	0.712665787
6	204558.02	39.45106659	0.448879896	6738.206448	74.44770063	0.743861757
7	192133.686	40.64587032	0.461503772	6161.745706	76.79215509	0.767194595
8	186214.548	41.48843238	0.471391838	5764.224674	78.29125338	0.781951829
9	176442.984	42.16473638	0.480101963	5414.526853	79.70018034	0.796398623
10	168389.232	42.89739905	0.487405242	5211.642409	80.61316501	0.805302846
11	159367.511	43.58215685	0.49469893	4985.338286	81.43597836	0.813601648
12	152099.616	44.18801251	0.501599348	4796.227293	82.1122633	0.820554683
13	147200.289	44.96576211	0.509431313	4614.106379	82.81109107	0.827653727
14	144353.454	45.47017218	0.514349581	4475.266947	83.44229035	0.833820707
15	139758.094	46.09575337	0.520900267	4356.63196	84.0509468	0.839842722

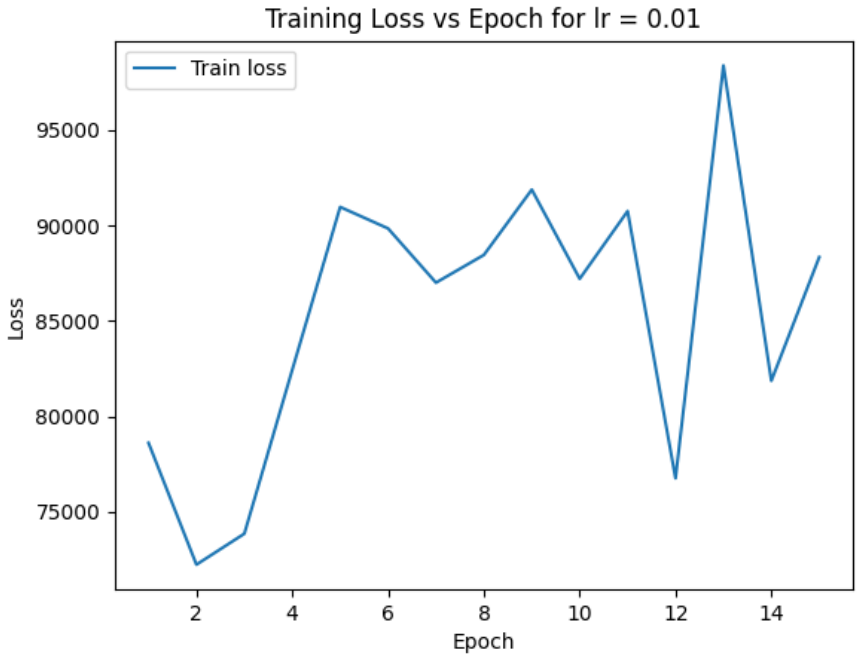
3.2 Table for Learning Rate = 0.0001

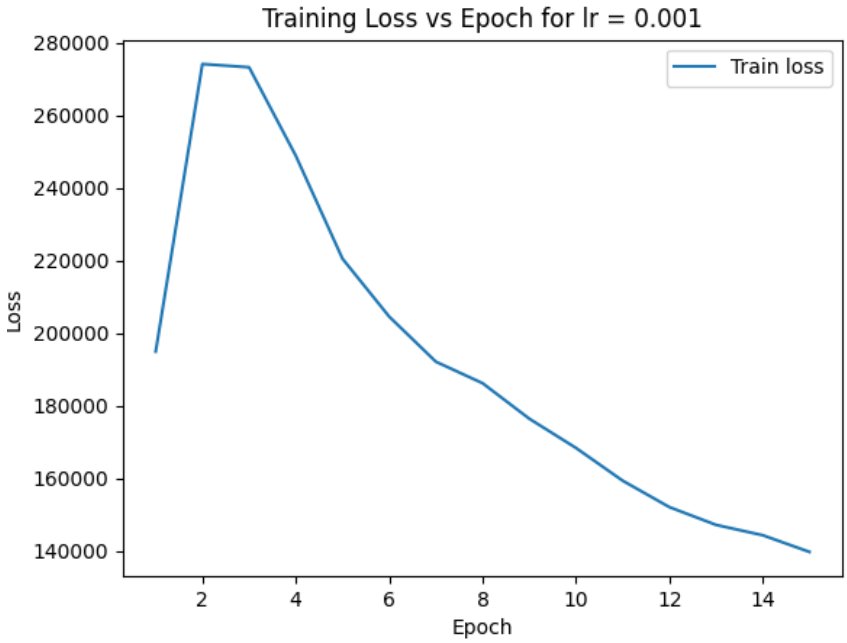
Epoch	Train Loss	Training Accuracy	Training F1 Score	Validation Loss	Validation Accuracy	Validation F1 Score
1	81341.89025	13.04984924	0.080748641	20380.21455	12.29711452	0.075364664
2	80947.152	14.9322287	0.104042116	20290.55969	13.52569883	0.094486201
3	80448.84237	16.96395863	0.119504554	20175.61528	15.61091073	0.109367018
4	79823.38749	19.63535943	0.149505697	20017.6912	18.15825068	0.13436366
5	78983.27293	22.88725449	0.184677764	19810.48641	20.75067628	0.161196458
6	77966.61919	25.95316595	0.215337394	19538.60833	24.05320108	0.194268809
7	76714.02586	28.06661594	0.237084411	19182.32655	26.58926961	0.218499855
8	75319.34141	29.38259081	0.253103125	18745.05963	29.0238954	0.243288415
9	74128.00037	29.10361541	0.258755184	18230.33154	31.51487827	0.271999932
10	73597.97024	27.58474934	0.257015084	17640.21187	34.4116321	0.314006754
11	75147.10549	25.47975315	0.248967856	16991.19984	37.63525699	0.357919569
12	79737.60578	24.95561755	0.253376518	16328.23062	40.39675383	0.393794474
13	87720.91948	25.26559022	0.26399599	15693.82718	43.01172227	0.424533313
14	98666.23122	25.93344041	0.27617765	15124.37278	45.37871957	0.450868916
15	109723.2976	26.58156508	0.28617741	14624.73778	47.26104599	0.47104648

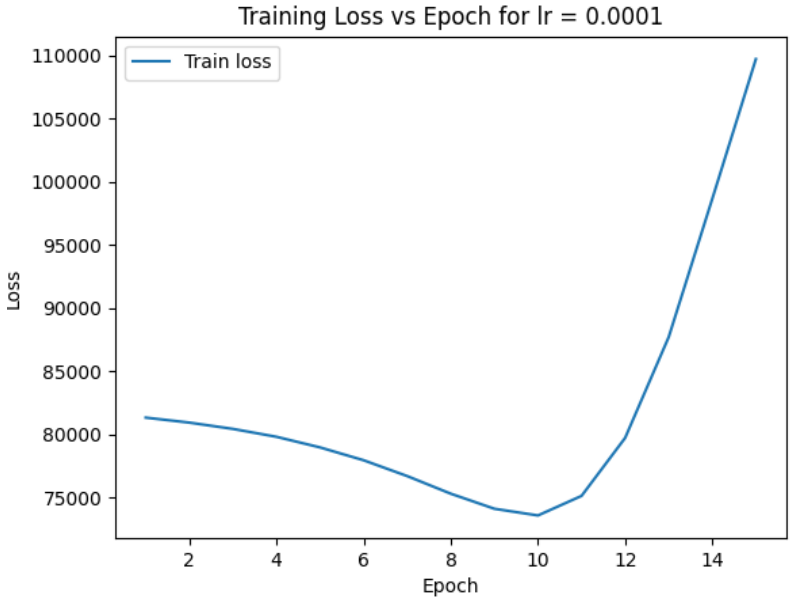
Selected best model: learning_rate = 0.01

4 Graphs

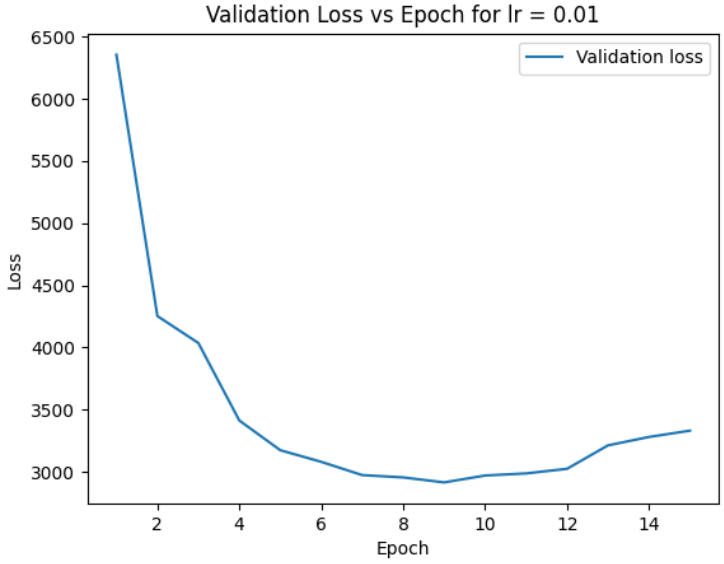
4.1 Training Loss

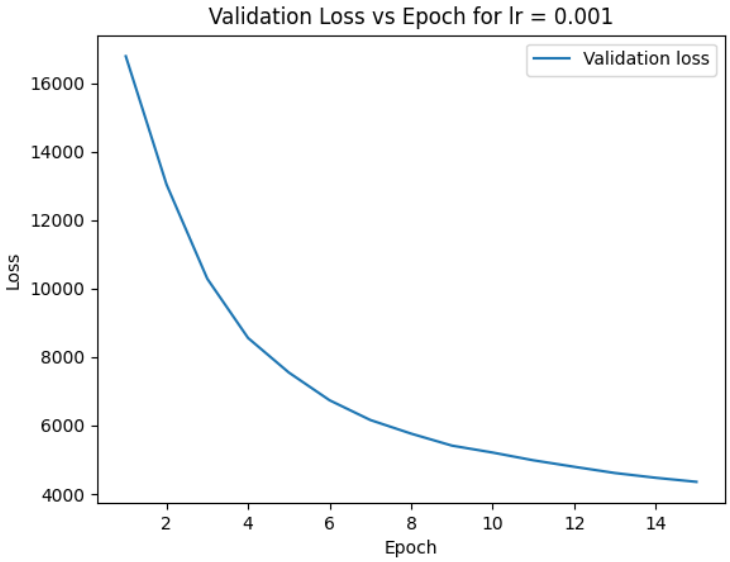


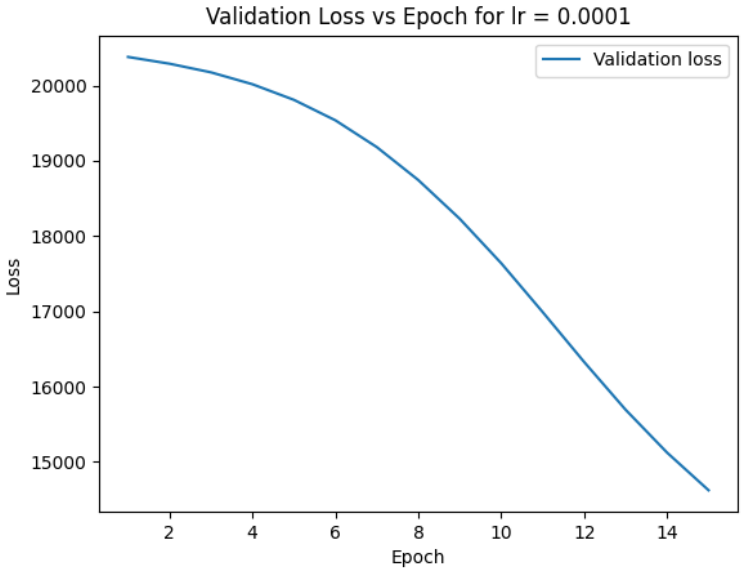




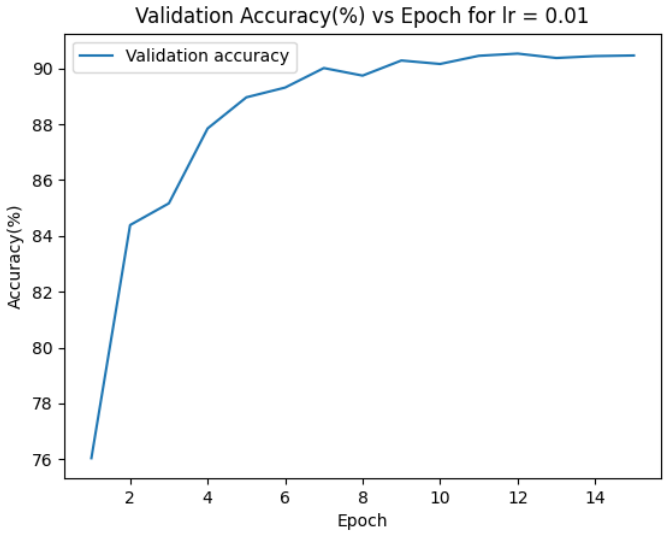
4.2 Validation Loss

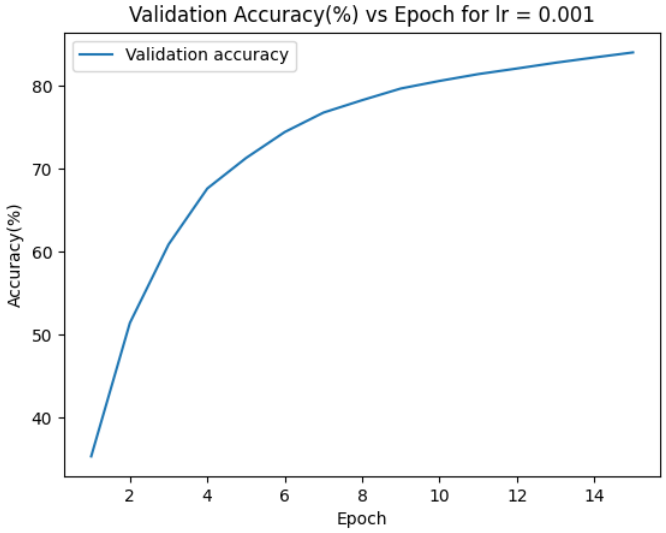


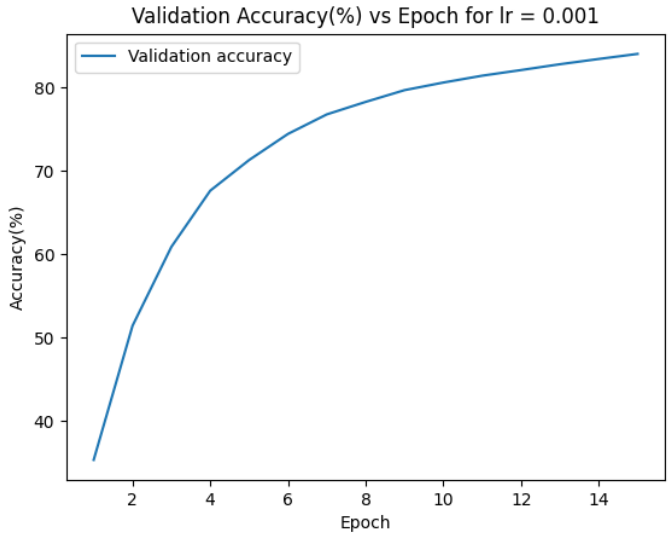




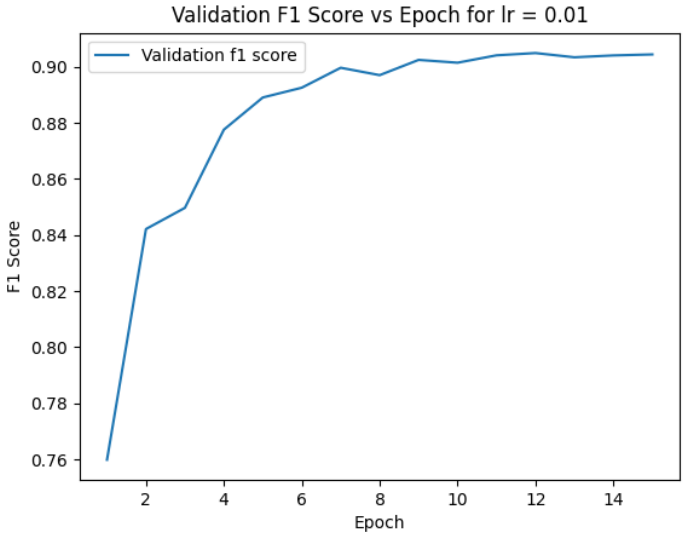
4.3 Validation Accuracy

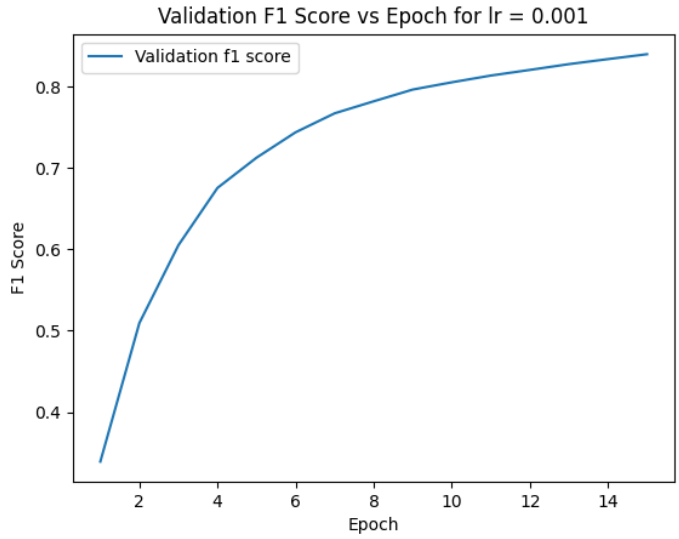

Validation Accuracy(%) vs Epoch for lr = 0.01
Validation accuracy
Accuracy(%)
Epoch
90
88
86
84
82
80
78
76
2
4
6
8
10
12
14

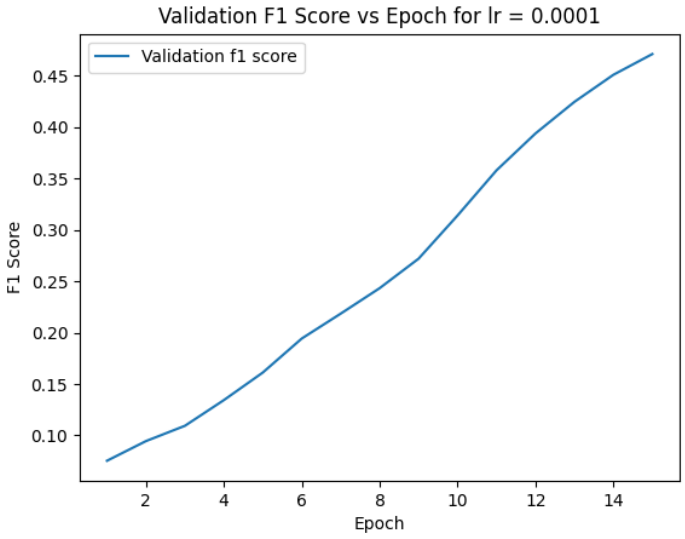

Validation Accuracy(%) vs Epoch for lr = 0.001
Validation accuracy
Accuracy(%)
Epoch
80
70
60
50
40
2
4
6
8
10
12
14



4.4 Validation Macro F1




Validation F1 Score vs Epoch for lr = 0.001
Validation f1 score
F1 Score
Epoch
0.4
0.5
0.6
0.7
0.8
2
4
6
8
10
12
14


Validation F1 Score vs Epoch for lr = 0.0001
Validation f1 score
F1 Score
Epoch
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
2
4
6
8
10
12
14

Discussion

1. The largest learning rate was selected as the best. We know, large learning rate means, faster convergence. There are downfalls of this like converging to a suboptimal solution. But as the number of epochs was very little, this performed the best.
2. The model was very slow at the beginning as there were nested loop. Then, vectorization was introduced to make it faster. Credit goes to a blog, and a stackoverflow answer. Without going through and implementing them it would be very difficult for me to make the model run faster.
3. Less epochs were used to train due to limited time.
4. More preprocessing, and applying latest models could have improved the result.