Project Report

Phase One:

For phase one we used the KNN Classifier and Support Vector Classifier(SVC) to classify images as “Bush vs Not Bush” and “Williams vs Not Williams”.

With the KNN classifier, when K is small we are restraining the region of a given prediction and forcing our classifier to be “more blind” to the overall distribution which means when the value for K is small, the classifier will have less bias and more variance. For our dataset, since we only have 530 images for Bush and 52 images from williams in the dataset of 13233 images the classifier is giving very low overall scores.

With SVC, if the value for penalty parameter C of the error term is between 10^2 - 10^5 and if the value for kernel coefficient gamma is taken relatively small, I am getting better F1 score for the classifier. Smaller value for gamma implies that we have low bias and high variance.

Following are my observations and best results for KNN and SVC classifiers:

KNN Classifier Results for Bush Dataset:

Classifier	KNeighborsClassifier			
Parameters	n_neighbors=1			
	result1	result2	result3	mean result
fit_time	7.11005497	7.10321999	6.82862616	7.01396704
score_time	985.9514918	986.3022468	986.674022	986.3092535
test_f1	0.13592233	0.13192612	0.17827298	0.1487071433
test_precision	0.15909091	0.12376238	0.17486339	0.1525722267
test_recall	0.11864407	0.14124294	0.18181818	0.1472350633
Classifier	KNeighborsClassifier			
Parameters	n_neighbors=3			
	result1	result2	result3	mean result
fit_time	7.72038698	7.68402505	7.49912	7.634510677
score_time	1013.852555	1016.225471	1015.840445	1015.306157
test_f1	0.04040404	0.10810811	0.08450704	0.07767306333
test_precision	0.19047619	0.26666667	0.24324324	0.2334620333
test_recall	0.02259887	0.06779661	0.05113636	0.04717728
Classifier	KNeighborsClassifier			
Parameters	n_neighbors=5			
	result1	result2	result3	mean result
fit_time	7.53382301	7.5316391	7.12480307	7.39675506
score_time	997.5689421	998.868979	999.1303649	998.522762
test_f1	0.02197802	0.06185567	0.04278075	0.04220481333
test_precision	0.4	0.35294118	0.36363636	0.3721925133
test_recall	0.01129944	0.03389831	0.02272727	0.02264167333

KNN Classifier Results for Williams Dataset:

Classifier	KNeighborsClassifier			
Parameters	n_neighbors=1			
	result1	result2	result3	mean result
fit_time	6.54868889	6.62668896	6.66906905	6.614815633
score_time	932.6906052	931.8940342	932.106024	932.2302211
test_f1	0.17391304	0.33333333	0.19047619	0.2325741867
test_precision	0.4	0.57142857	0.5	0.49047619
test_recall	0.11111111	0.23529412	0.11764706	0.1546840967
Classifier	KNeighborsClassifier			
Parameters	n_neighbors=3			
	result1	result2	result3	mean result
fit_time	18.00533891	17.43023491	16.72769284	17.38775555
score_time	945.728673	944.1679122	944.7746942	944.8904265
test_f1	0	0	0	0
test_precision	0	0	0	0
test_recall	0	0	0	0
Classifier	KNeighborsClassifier			
Parameters	n_neighbors=5			
	result1	result2	result3	mean result
fit_time	8.25286794	8.65910506	8.32191896	8.41129732
score_time	1007.936384	1006.054879	1004.968208	1006.319824
test_f1	0	0	0	0
test_precision	0	0	0	0
test_recall	0	0	0	0

Best (in terms of mean F1) SVC result I got for Bush Dataset:

Parameters	C=200	kernel=rbf	gamma='auto'	
	result1	result2	result3	mean result
fit_time	61.84511328	58.40753961	62.86394739	61.03886676
score_time	76.34336472	73.22346711	76.6356833	75.40083838
test_f1	0.65780731	0.63013699	0.64451827	0.64415419
test_precision	0.7983871	0.8	0.776	0.7914623667
test_recall	0.55932203	0.51977401	0.55113636	0.5434108

Best (in terms of mean F1) SVC result I got for Williams Dataset:

Parameters	C=1000	kernel=rbf	gamma='auto'	
	result1	result2	result3	mean result
fit_time	10.72223973	11.82295012	12.22219348	11.58912778
score_time	10.58737326	11.08349013	11.21315384	10.96133908
test_f1	0.55172414	0.68965517	0.71428571	0.65188834
test_precision	0.72727273	0.83333333	0.90909091	0.8232323233
test_recall	0.44444444	0.58823529	0.58823529	0.5403050067

Phase Two:

In Phase Two we are evaluating performance of the KNN classifier given various Principal Component Analysis (PCA) parameters. Principal Component Analysis is to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent. I tried PCA parameters : 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096.

Following are the best evaluations and the parameters for which I got best scores for KNN classifier with Principal Component Analysis parameter values:

Best result for KNeighborsClassifier for Bush Dataset:

PCA parameters	32
KNeighborsClassifier parameters	1
Mean F1	0.1570943378

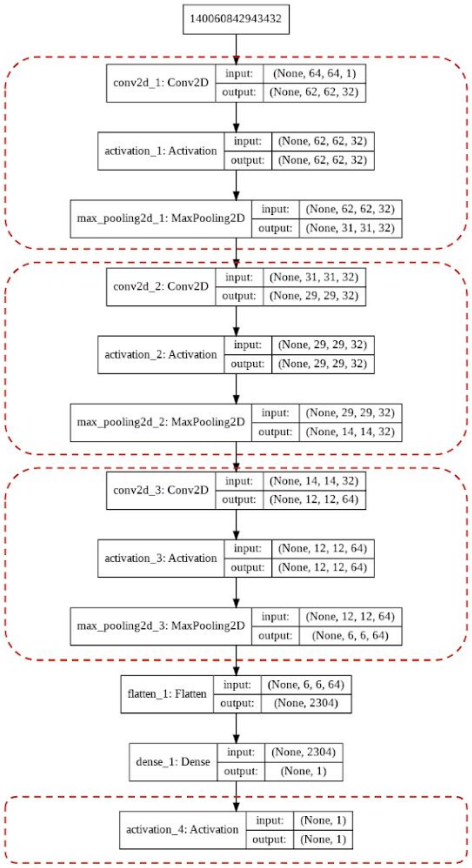
Best result for KNeighborsClassifier for Williams Dataset:

PCA parameters	16
KNeighborsClassifier parameters	1
Mean F1	0.2555485893

Phase Three:

In this phase, we built a deep learning convolutional neural network using a few hidden layers to built a classification model. Since the dataset for positive cases, i.e images for Bush and Serena Williams is very small if I added more than 2 hidden layers to my neural network the score did worse. I also observed that for my neural network, if I increased the number of epochs the accuracy got better and the loss decreased but only upto a certain number of epochs. After that the value for accuracy and loss pretty much remained constant.

Following is the model that I used for Bush Dataset:



Convolutional Neural Network - Bush

Layer 1: Activation - relu

Layer 2: Activation - tanh

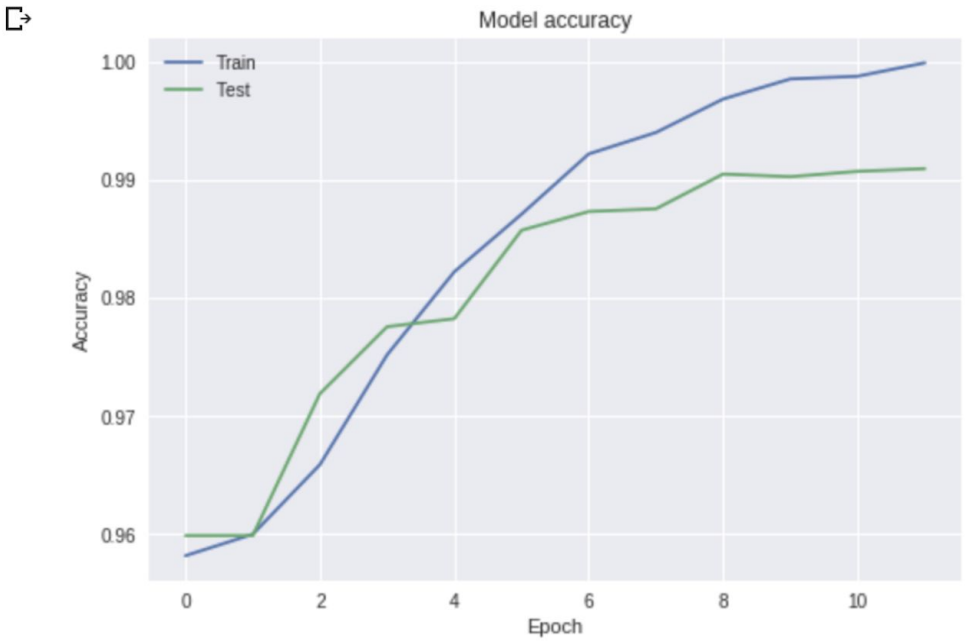
Layer 3: Activation - relu

Layer 4: Activation - sigmoid

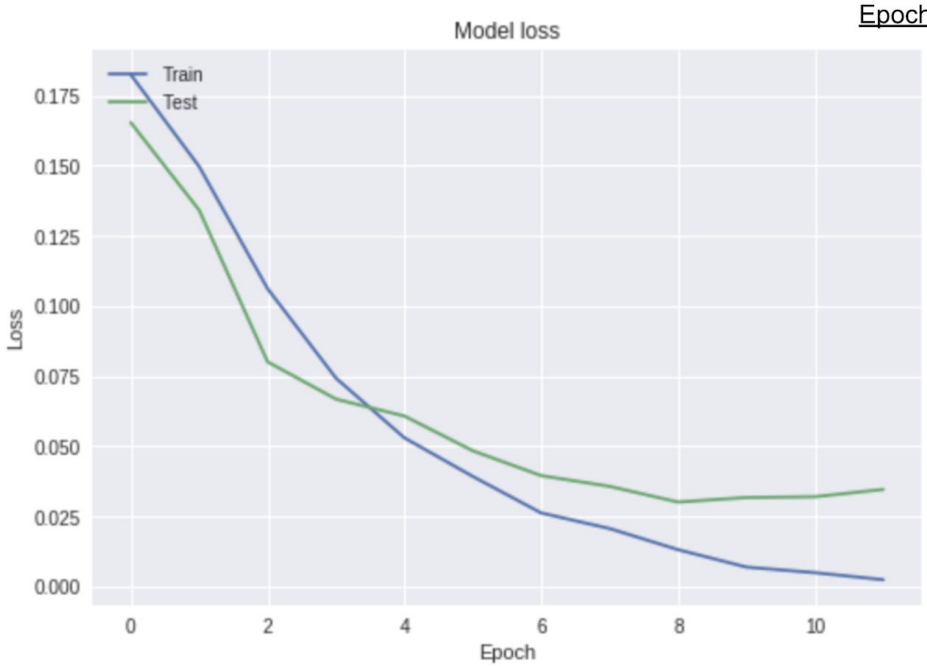
Best F1 score for Bush: 0.8765432098765431

Following are the graphs for Accuracy vs Epoch and Loss vs Epoch for Bush Dataset:

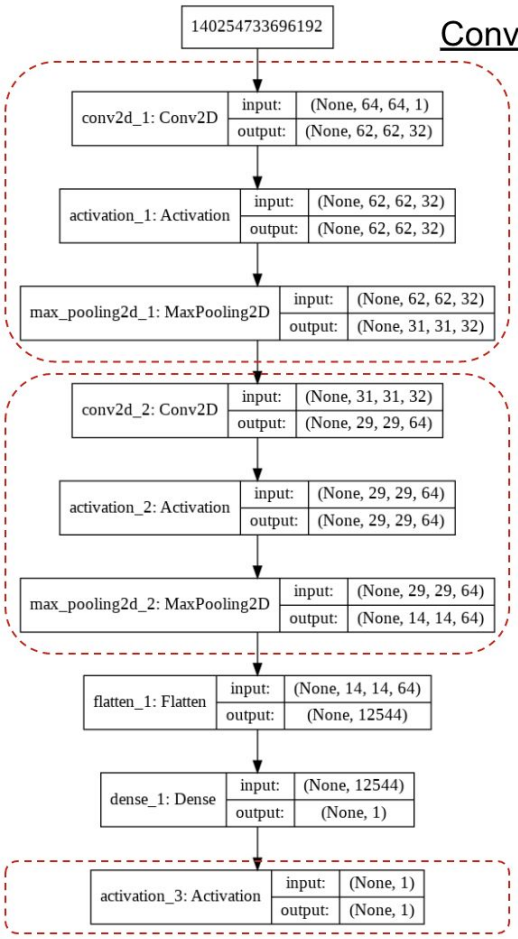
Epoch vs Accuracy - Bush



Epoch vs Loss - Bush



Following is the model that I used for Williams Dataset:



Convolutional Neural Network - Williams

Layer 1: Activation - tanh

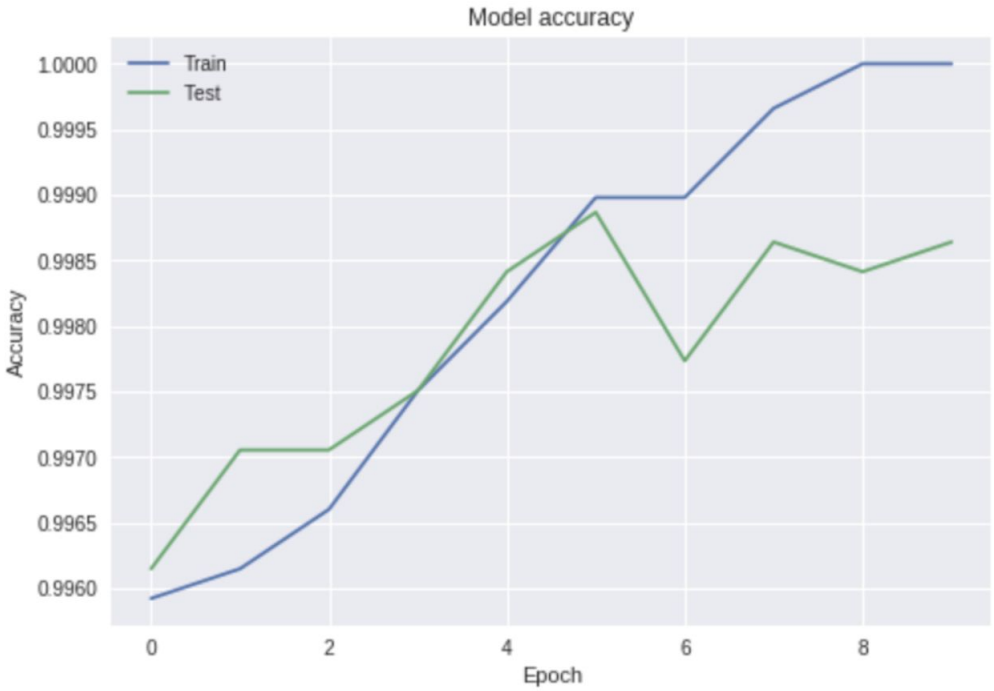
Layer 2: Activation - relu

Layer 3: Activation - sigmoid

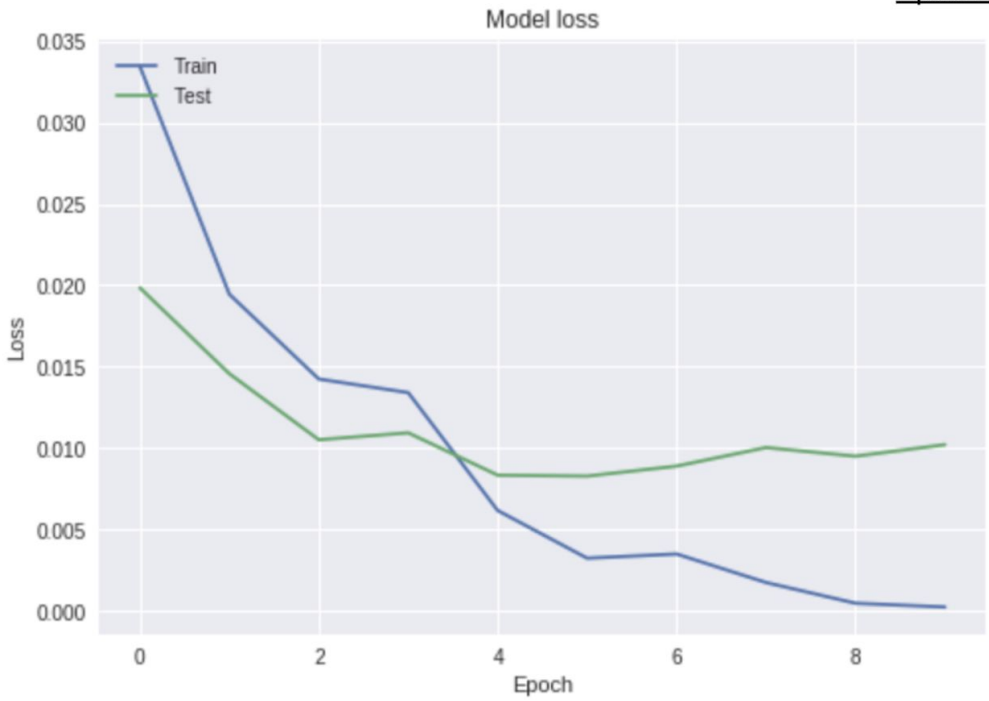
F1 score for Williams: 0.8000000000000002

Following are the graphs for Accuracy vs Epoch and Loss vs Epoch for Williams Dataset:

Epoch vs Accuracy - Williams



Epoch vs Loss - Williams



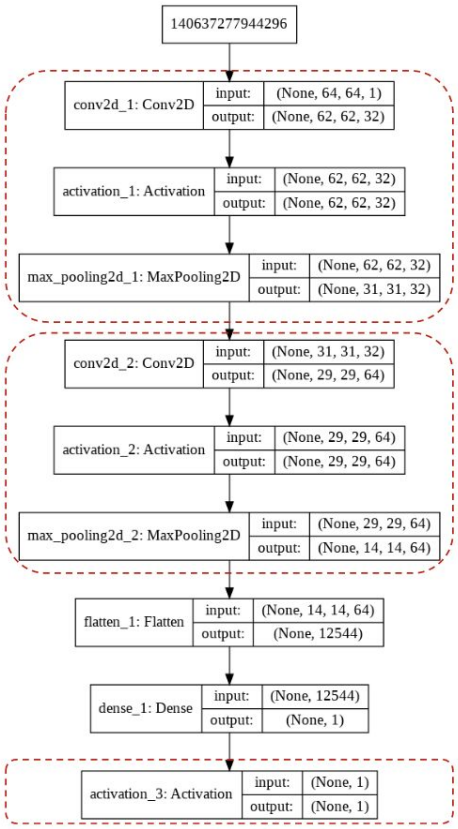
Phase Four:

In phase 4 we performed simple transfer learning. I trained my convolutional neural network model on a dataset to classify cats vs dogs images, used this pre trained model to then further train on the “bush vs others” and “williams vs others” dataset to see how the F1 score gets affected if we classify using transfer learning.

For the initial model I used one hidden layer with “relu” as the activation function, the output layer with “sigmoid” as the activation function and input layer with “relu” as the activation function. After pre training the model I trained the model with bush and williams dataset separately to predict F1. I tried running multiple iterations to pre training the model on “cats vs dogs” dataset and then further training the model on “bush vs others” or “williams vs others” dataset. I observed that if my pre trained model did as good as ~90% on the training dataset for “cats vs dogs” it performed better classification in the step 2 of the training. If I further tried to improve my F1 score for the pre trained model it performed bad for “bush vs others” and “williams vs others” transfer learning classification. I also observed that for my neural network, if I increased the number of epochs the accuracy got better and the loss decreased but only upto a certain number of epochs. After that the value for accuracy and loss pretty much remained constant.

Dataset URL: https://www.kaggle.com/gurjarboy/cnn-basic-cat-vs-dog/data

Following is the model for both bush and williams dataset:



Convolutional Neural Network

Layer 1: Activation - relu

Layer 2: Activation - relu

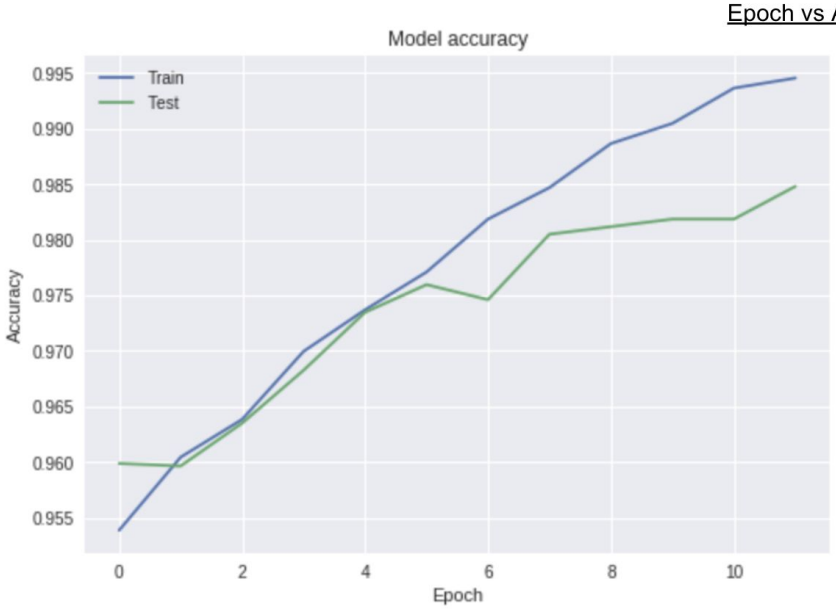
Layer 3: Activation - sigmoid

F1 Score:

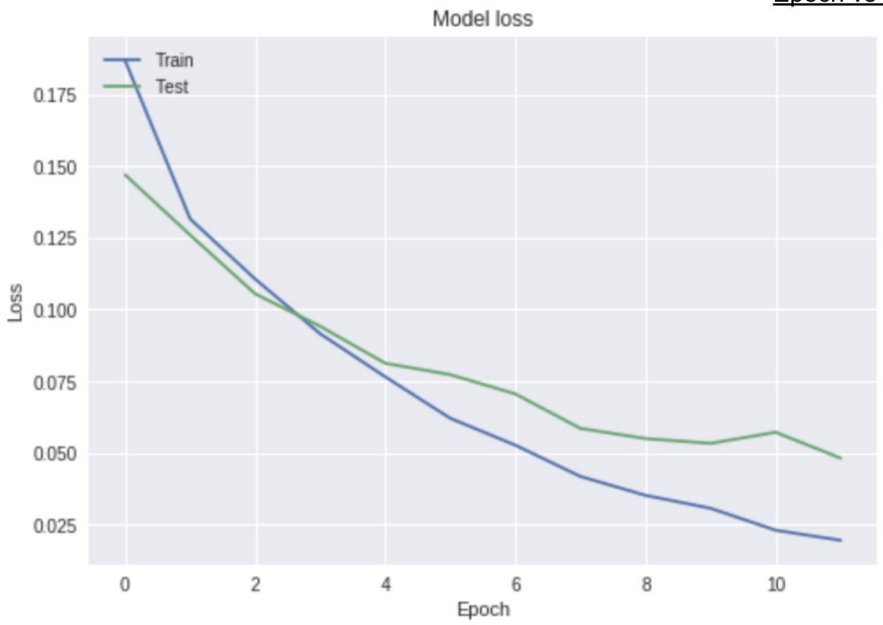
	Bush	Williams
F1_Train	0.9842632331902718	1
F1_Test	0.7975830815709971	0.7096774194

Following are the graphs for Accuracy vs Epoch and Loss vs Epoch for Bush Dataset:

Epoch vs Accuracy - Bush

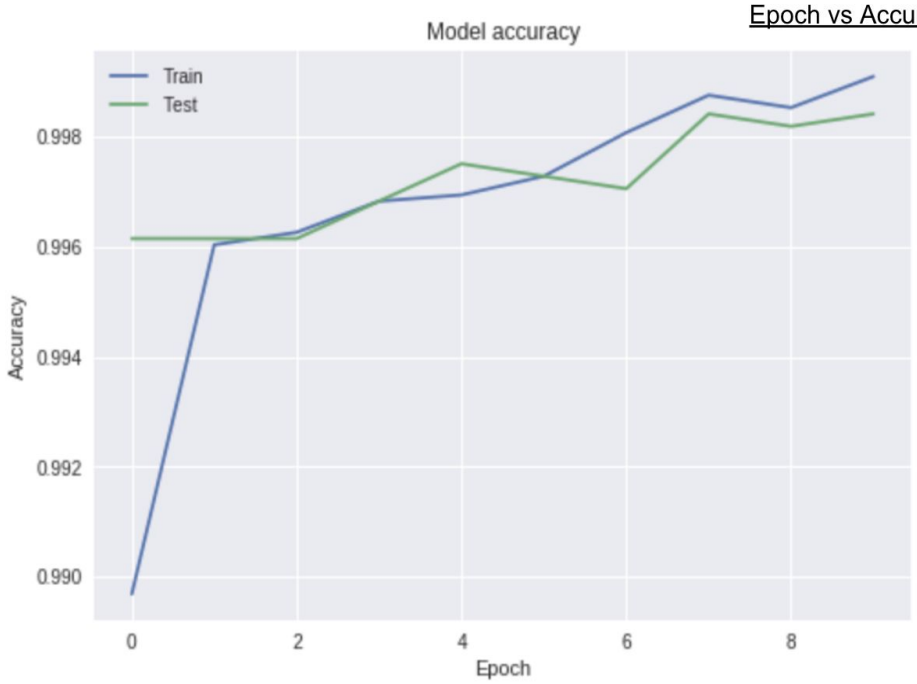


Epoch vs Loss - Bush



Following are the graphs for Accuracy vs Epoch and Loss vs Epoch for Williams Dataset:

Epoch vs Accuracy - Williams



Epoch vs Loss - Williams