Assignment 3
Building Neural Networks and CNN

Ram Chandra Bhavirisetty
Sanjana Veesam

"We certify that the code and data in this assignment were generated independently, using only the tools and resources defined in the course and that I (we) did not receive any external help, coaching or contributions during the production of this work."

PART-1

1. Provide brief details about the nature of your dataset. What is it about? What type of data are we encountering? How many entries and variables does the dataset comprise? Provide the main statistics about the entries of the dataset.

- Our income dataset has 6 variables of type ‘integer’ and 9 categorical variables.
- It's a combination of quantitative and qualitative data, comparatively we have more qualitative data.
- There are 32561 entries and 15 variables in this dataset.
- We can get the main statistics of the data from describe().
- From these statistics we can say that this dataset has people of age group 17 to 90.
- On an average, hours per week worked is 40.

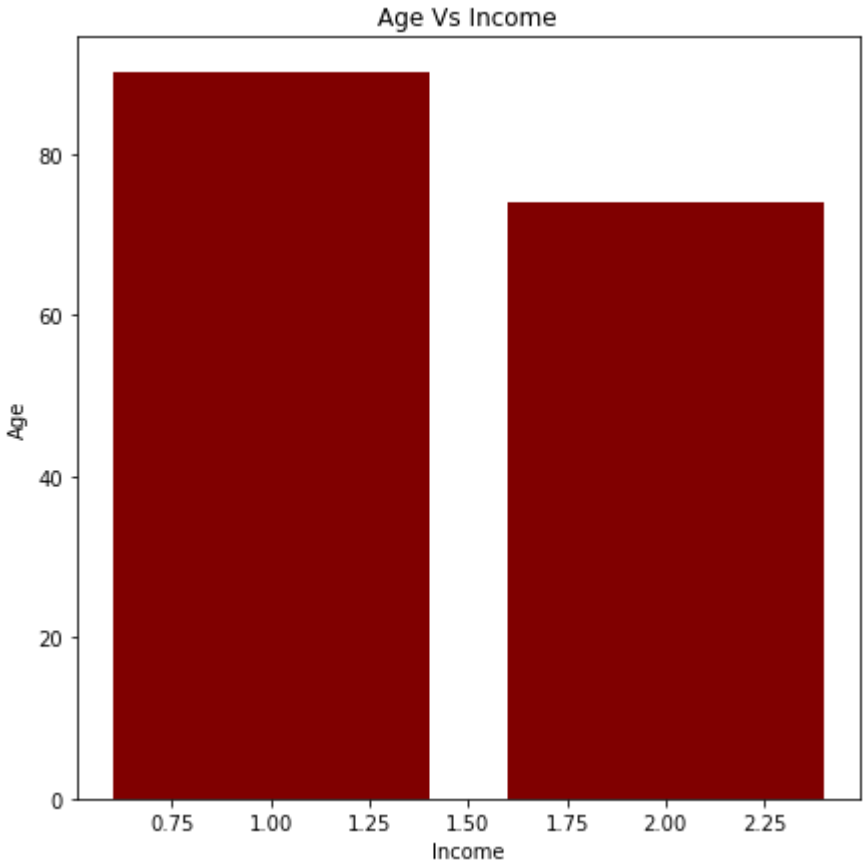
```
: data.describe()
```

:

	age	fnlwgt	education.num	capital.gain	capital.loss	hours.per.week
count	32561.000000	3.256100e+04	32561.000000	32561.000000	32561.000000	32561.000000
mean	38.581647	1.897784e+05	10.080679	1077.648844	87.303830	40.437456
std	13.640433	1.055500e+05	2.572720	7385.292085	402.960219	12.347429
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.178270e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.783560e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.370510e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.484705e+06	16.000000	99999.000000	4356.000000	99.000000

2. Provide at least 3 visualisation graphs with short descriptions for each graph.

Age Vs Income :

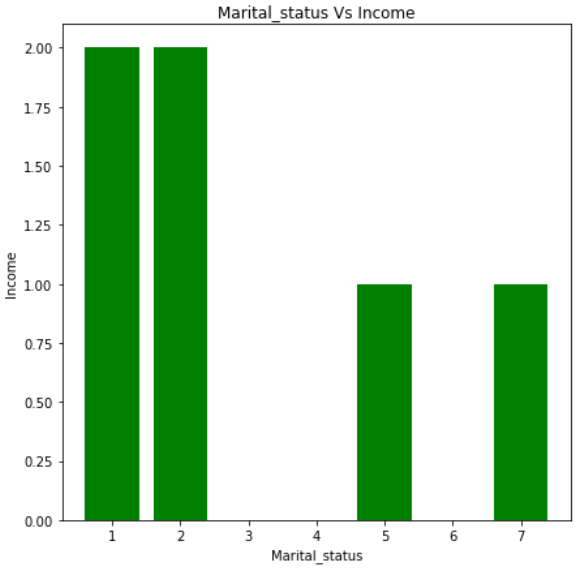


By this graph we can say that, majority of people have income less than 50k of all age groups.

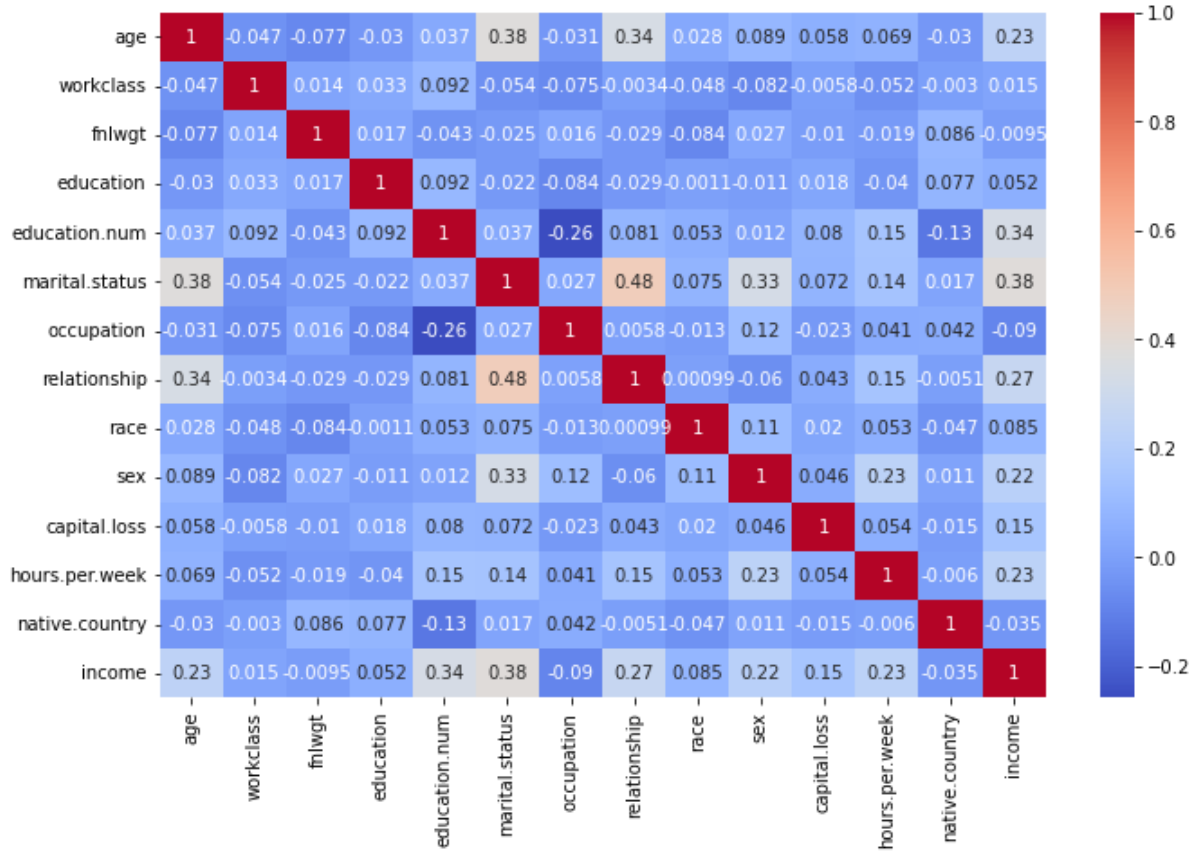
Income Vs Marital Status:

By this graph we can conclude that
'Never-married': 1,
'Divorced': 2,
'Married-AF-spouse': 3,
'Separated': 4,
'Married-civ-spouse': 5,
'Widowed': 6,
'Married-spouse-absent': 7

Average of people with their marital status as ‘Never-Married’ and ‘Divorced’ has highest income compared to others.



Heat Map:



Through this heat-map we can get the features which are positively correlated with the target feature**.**

3. For the preprocessing part, discuss if you use any preprocessing tools that help to increase the accuracy of your model.

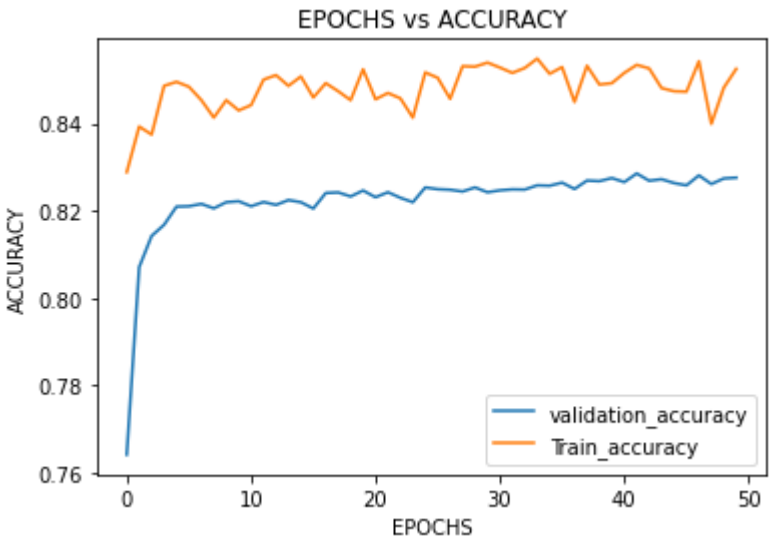
Preprocessing :

- Dropping the capital.gain column
- Finding the names of the columns which contain '?' and replacing it with the mode of that column.
- Replacing categorical data with numericals.
- Data normalisation (Min-Max scalar).

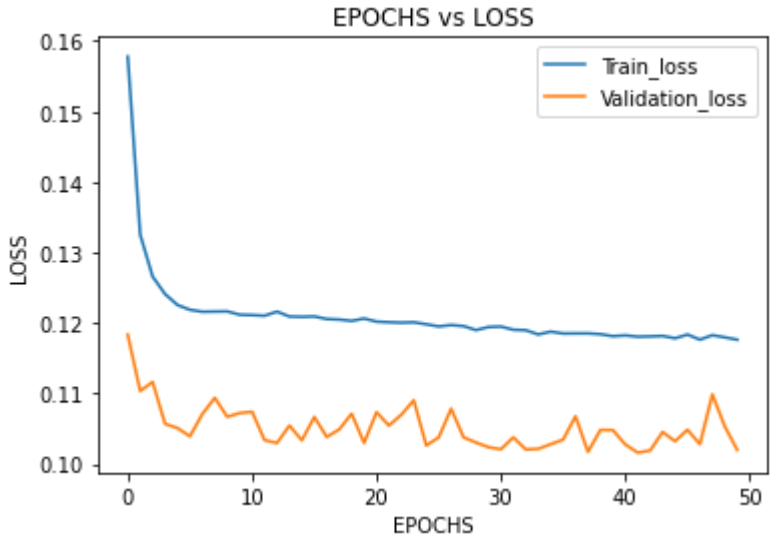
4. Provide the architecture structure of your NN.

Using Sequential model from the keras library to create NN Architecture, it consists of 1 input layer, 3 hidden layers and 1 Output layer. Linear, relu and sigmod are the activation functions used for this architecture. Adam optimizer.

5. Provide graphs that compare test and training accuracy on the same plot, test and training loss on the same plot. Thus in total there are two graphs with clear labelling.



With the increasing number of epochs, accuracy is increasing. Validation accuracy is more compared to training accuracy.



With the increasing number of epochs, loss is decreasing. Validation loss is less compared to training loss.

PART-2

1. Include all 3 tables with different NN setups.

	setup-1	Accuracy	setup-2	Accuracy	setup-3	Accuracy
Dropout	0.2	85.167	0.5	85.09571	0.75	84.75
Optimizer	“Adam”		“Adam”		“Adam”	
Activation function	‘Linear’ ‘Relu’, ‘sigmoid’		‘Linear’, ‘Relu’ ‘sigmoid’		‘Linear’, ‘Relu’ ‘sigmoid’	
Intializer	random_normal		random_normal		random_normal	

	setup-1	Accuracy	setup-2	Accuracy	setup-3	Accuracy
Dropout	0.2	85.095701	0.2	84.000	0.2	85.003
Optimizer	“Adam”		“sgd”		“rmsprop”	
Activation function	‘Linear’ ‘Relu’, ‘sigmoid’		‘Linear’, ‘Relu’ ‘sigmoid’		‘Linear’, ‘Relu’ ‘sigmoid’	

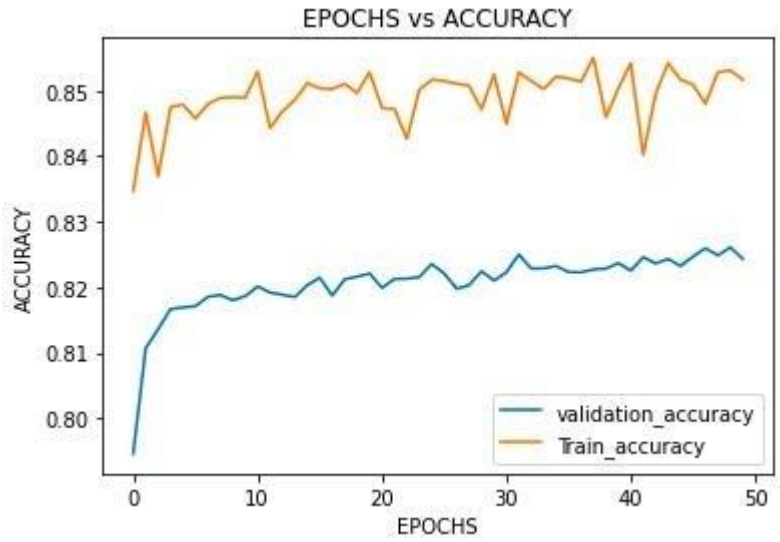
Intializer	random_normal		random_normal		random_normal	

	setup-1	**Accuracy**	**setup-2**	**Accuracy**	**setup-3**	**Accuracy**
Dropout	0.2	84.45	0.2	80.74	0.2	84.249
Optimizer	“Adam”		“Adam”		“Adam”	
Activation function	‘Linear’ ‘Relu’, ‘sigmoid’		‘Linear’, ‘Relu’ ‘sigmoid’		‘Linear’, ‘Relu’ ‘sigmoid’	
Intializer	random_normal		zeroes		random_uniform	

In the first setup , maximum accuracy occurs at dropout=0.2, in this case we take dropouts as 0.2 for the remaining setups.
Maximum accuracy occurs when :
dropout=0.2
optimizer=adam
initializer=random_normal

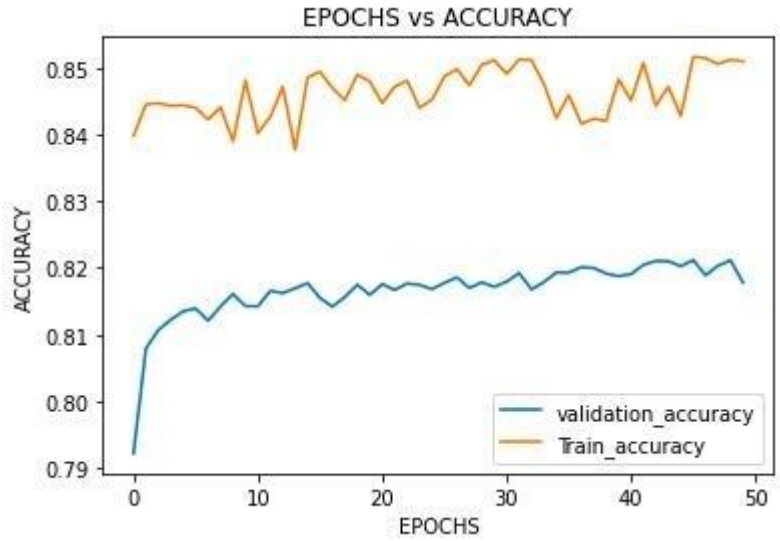
2. Provide graphs that compare test and training accuracy on the same plot for all your setups and add a short description for each graph.

Setup 1:



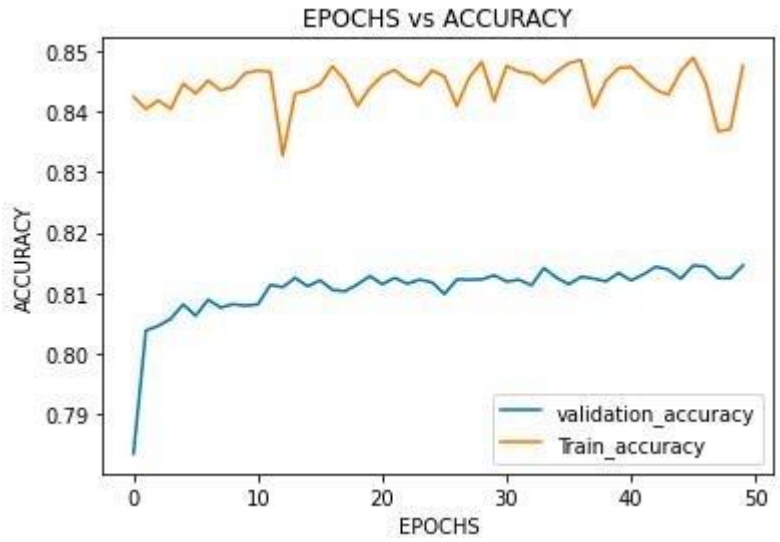
In this setup, Train accuracy is higher than validation accuracy. There is a huge accuracy variation for both the datasets.Train accuracy is highly fluctuating throughout the graph.

Setup 2:



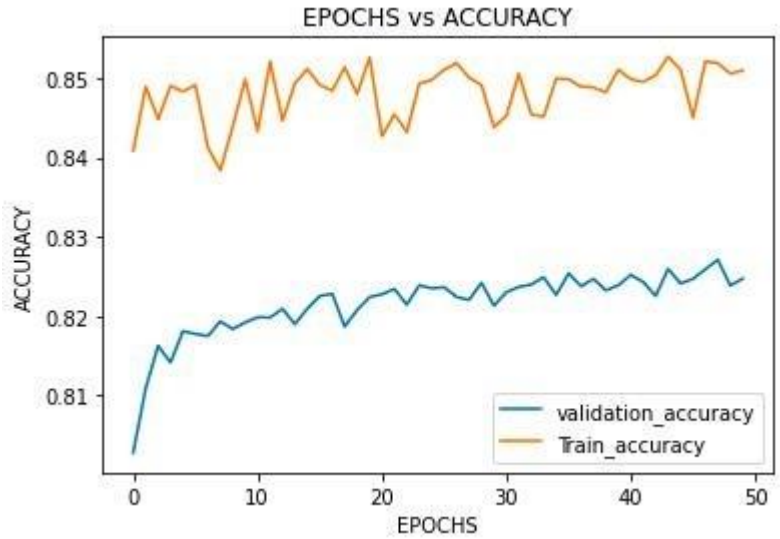
In this setup, Train accuracy is higher than validation accuracy. There is a huge accuracy variation for both the datasets.Train accuracy is highly fluctuating throughout the graph, validation accuracy has a slight variation.

Setup 3:



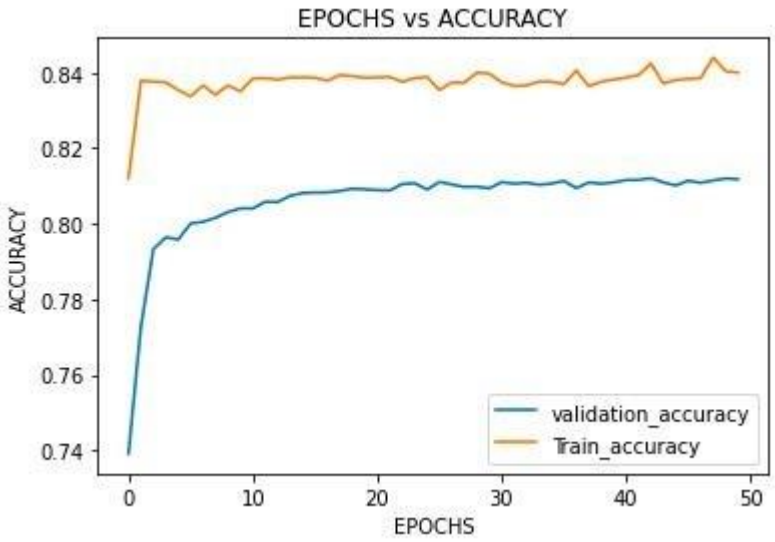
In this setup, Train accuracy is higher than validation accuracy. We have the lowest train accuracy when the number of epochs is close to 13 or 13.

Setup 4:



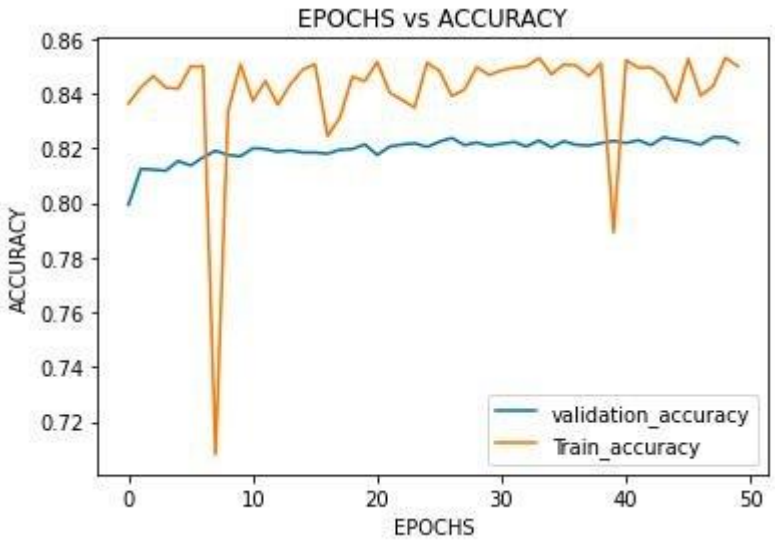
In this setup, Train accuracy is higher than validation accuracy. Lowest train accuracy is at epoch=10.

Setup 5:



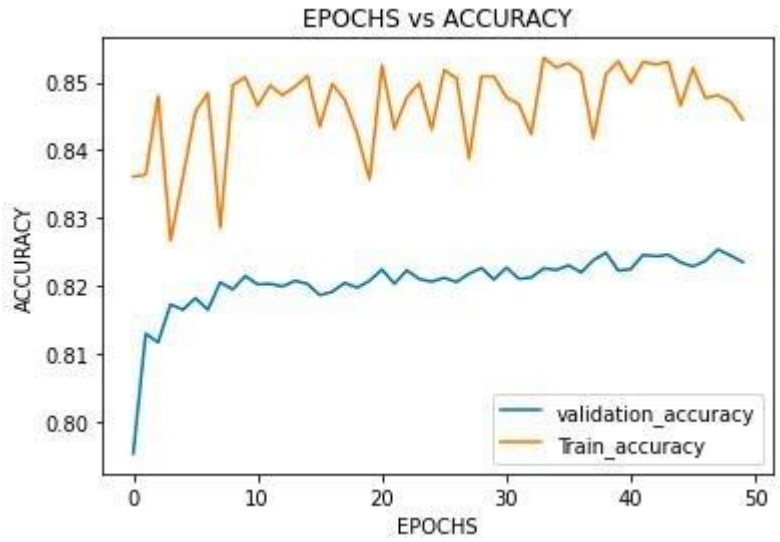
In this setup, Train accuracy is higher than validation accuracy. Graphs in this setup for train data are almost stable with slight variation.

Setup 6:



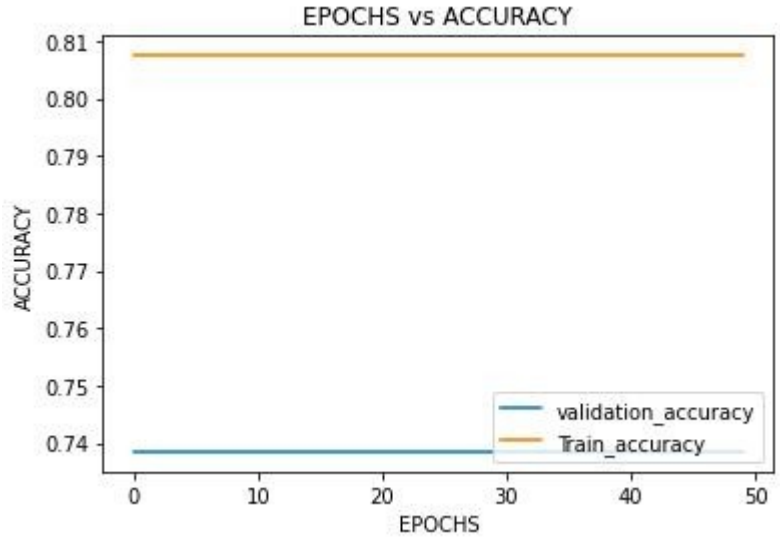
In this setup, Train accuracy dropped drastically at epoch=8 but validation accuracy is stable throughout the graph.

Setup 7:



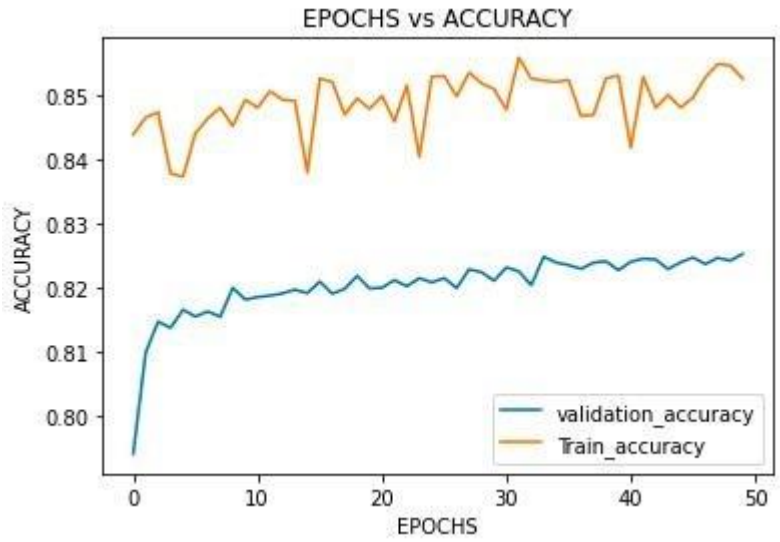
In this setup, Train accuracy dropped initially and fluctuated throughout the graph.

Setup 8:



In this setup, train and validation accuracies are stable throughout the graph, but has huge variation.

Setup 9:

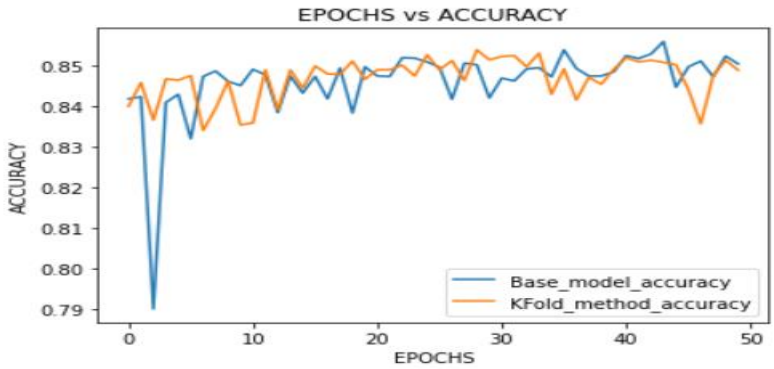


In this setup, boh the values are fluctuating.

3. Provide a detailed analysis and reasoning about the NN setups that you tried.

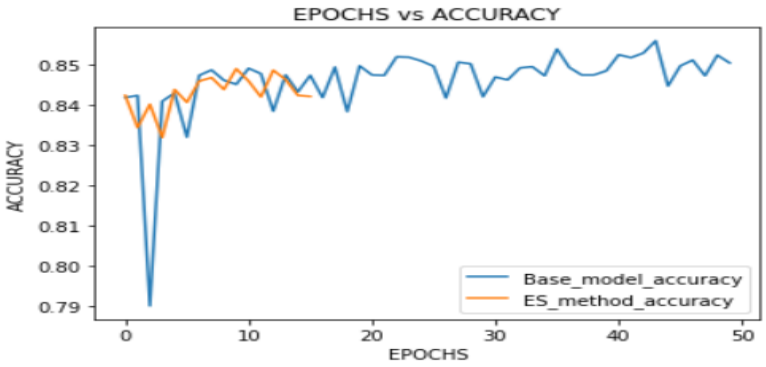
There are 9 setups in total.
These are slight variations of original architecture, among all the setups the accuracy occurred when :
Droupt= 0.2
optimizer=Adem
Intializer= Random.uniform

4. Briefly discuss all the methods you used that help to improve the accuracy or training time. Provide accurate graphs and your short descriptions.
Optimization methods we used are:
K-Fold



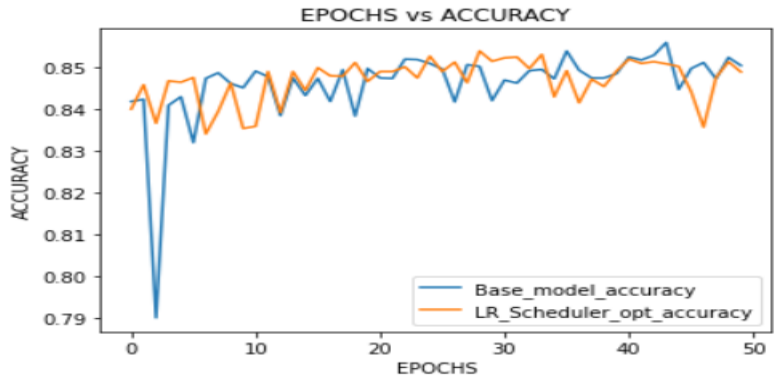
In this setup, Both Base model accuracy and KFold method accuracy has a its own ups and down but overall it's evident that the accuracy gradually increases though Basemodel drops to the least at a point and they both meet at several points.

Early Stopping



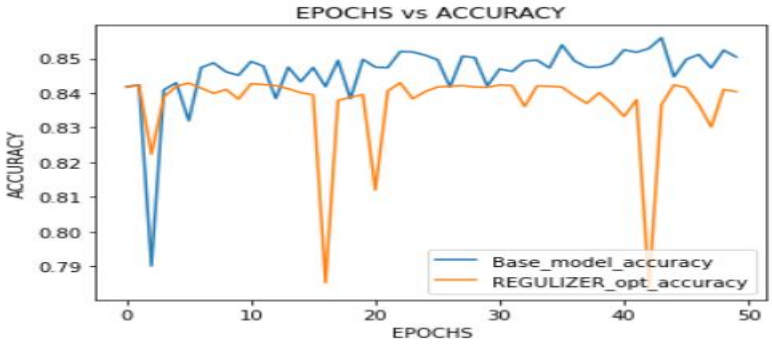
In this setup, Base model has a its own ups and down but overall it's evident that the accuracy gradually increases though it drops to the least at a point and they both meet at several points yet ES method accuracy stops after a point.

Learning rate decay scheduler



In this setup, Both Base model accuracy and LR_scheduler opt accuracy has a its own ups and down but overall it's evident that the accuracy gradually increases though Basemodel drops to the least at a point and they both meet at several points

Regularisation



In this setup, Both Base model accuracy and Regulizer opt accuracy has a its own ups and down but overall it's evident that the Base model accuracy gradually

increases though Both drops to the least at a lot of points and they both also meet at several points

PART-3

1. Provide brief details about the nature of your dataset. What is it about? What type of data are we encountering? How many entries and variables does the dataset comprise? Provide the main statistics about the entries of the dataset.

Training Images:

The Fashion Mnist dataset consists of 60,000 entries, each entry is described as an image with 784 pixels.

Entire data is numerical.

Training Labels:

There are 10 classes, each class describes a fashion entity.

Testing Images:

The Fashion Mnist dataset consists of 10,000 entries, each entry is described as an image with 784 pixels.

Entire data is numerical.

Testing Labels:

There are 10 classes, each class describes a fashion entity.

```
{0: 'T-shirt/top',
 1: 'Trouser',
 2: 'Pullover',
 3: 'Dress',
 4: 'Coat',
 5: 'Sandal',
 6: 'Shirt',
 7: 'Sneaker',
 8: 'Bag',
 9: 'Ankle boot'}
```

These are 10 classes.

	label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...
count	60000.000000	60000.000000	60000.000000	60000.000000	60000.000000	60000.000000	60000.000000	60000.000000	60000.000000	60000.000000	...
mean	4.500000	0.000900	0.006150	0.035333	0.101933	0.247967	0.411467	0.805767	2.198283	5.682000	...
std	2.872305	0.094689	0.271011	1.222324	2.452871	4.306912	5.836188	8.215169	14.093378	23.819481	...
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
25%	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
50%	4.500000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
75%	7.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
max	9.000000	16.000000	36.000000	226.000000	164.000000	227.000000	230.000000	224.000000	255.000000	254.000000	...

2. Provide at least 3 visualisation graphs with a short description for each graph.

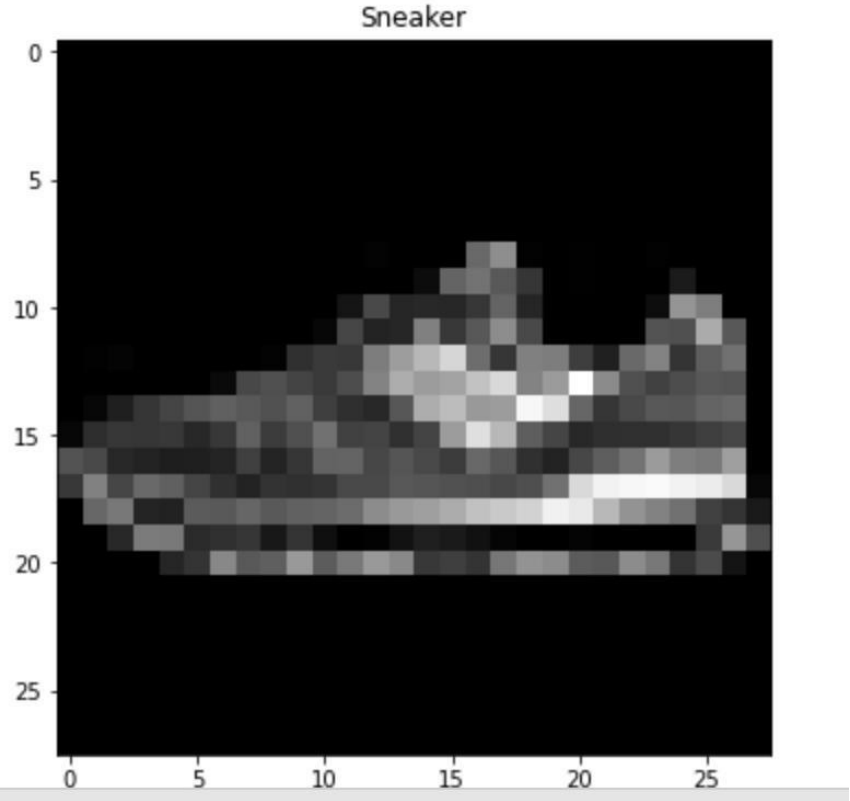
Graph 1:

Value obtained with actual data.

Graph 2:

With 93% of accuracy, the above values are predicted and labels marked in red are error values.

Graph 3:



This is the sample image with no label.

3. Provide the architecture structure of your CNN.

Layer 1: Convolutional layer
This consists of 64 filters with kernel size of 3 and it takes the input of shape 28x28x1 which is nothing but the reshape of all the images.
Layer 2: Max pooling 2D
This has a pool size of 3.

Layer 3: Flatten
This is the layer which flattens the input flow, these inputs can be used by fully connected layers.

Layer 2 and 3 are fully connected layers with 1024 and 512 nodes; these use relu as activation functions.

Output layer: It's a dense layer with nodes and uses softmax as an activation function.

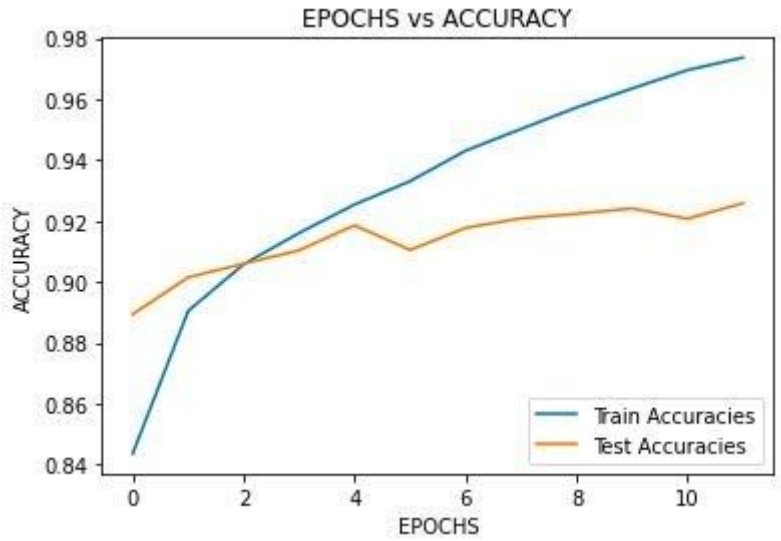
4. Discuss how the improvement tools work on CNN architectures.

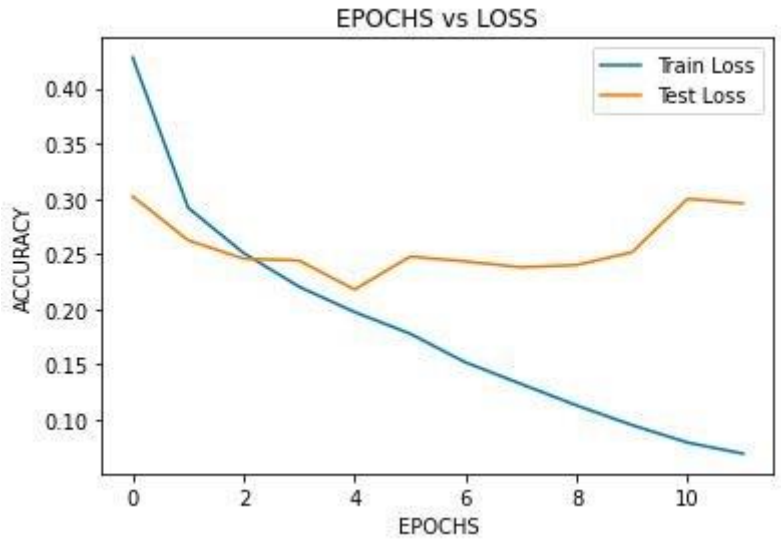
Improvement tools we have used for the above architecture are:

Dropout
Optimizer
Intializer
Kernel Size
Padding

Among these, accuracy gets highly affected with changing dropout values.

5. Provide graphs that compare test and training accuracy on the same plot, test and training loss on the same plot. Thus in total there are two graphs with clear labelling.





PART-4

1. Include all 3 tables with different CNN setups.

	setup-1	Accuracy	setup-2	Accuracy	setup-3	Accuracy
Dropout	0.2	92.65%	0.2	93.2%	0.2	87.65%
Optimizer	adam		adam		adam	
Activation function	Relu, softmax		Relu, softmax		Relu, softmax	
Intializer	random_u nifrom		random_u nifrom		random_u nifrom	
Kernel Size	3		2		1	
Padding	“same”		“same”		“same”	
Strides	1		1		1	

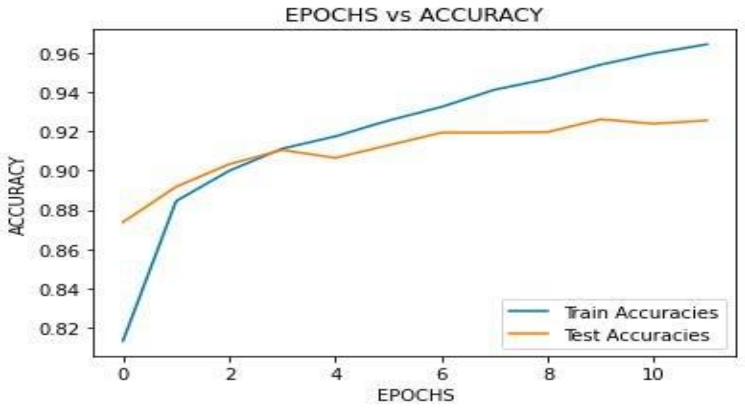
	setup-1	Accuracy	setup-2	Accuracy	setup-3	Accuracy
Dropout	0.2	92.8 %	0.5	92.3%	0.75	92.2 %
Optimizer	adam		adam		adam	
Activation function	Relu, softmax		Relu, softmax		Relu, softmax	
Intializer	random_u nifrom		random_u nifrom		random_u nifrom	
Kernel Size	1		1		1	
Padding	“same”		“same”		“same”	
Strides	1		1		1	

	setup-1	Accuracy	setup-2	Accuracy	setup-3	Accuracy
Dropout	0.2	88.6 %	0.2	67.75	0.2	88.2 %
Optimizer	adam		sgd		rmsprop	
Activation function	Relu, softmax		Relu, softmax		Relu, softmax	
Intializer	random_u nifrom		random_u nifrom		random_u nifrom	
Kernel Size	1		1		1	
Padding	“same”		“same”		“same”	
Strides	1		1		1	

	setup-1	Accuracy	setup-2	Accuracy	setup-3	Accuracy
Dropout	0.2	84.3 %	0.2	89.2 %	0.2	91.85 %
Optimizer	adam		adam		adam	
Activation function	Relu, softmax		Relu, softmax		Relu, softmax	
Intializer	random_u nifrom		random_u nifrom		random_u nifrom	
Kernel Size	1		1		1	
Padding	“same”		“same”		“same”	
Strides	3		2		1	

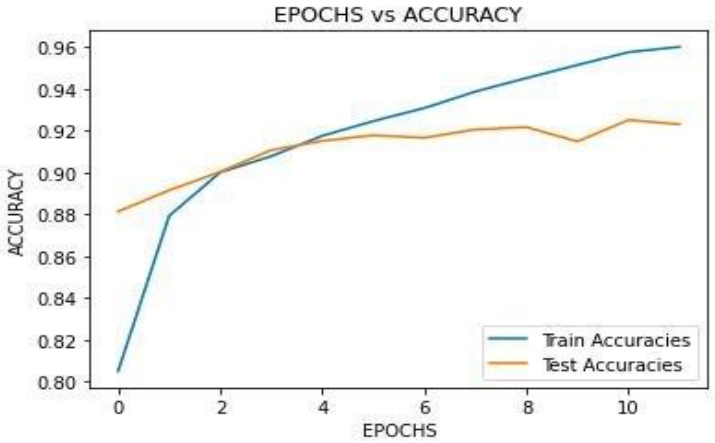
2. Provide graphs that compare test and training accuracy on the same plot for all your setups and add a short description for each graph.

Setup :1



In this setup, Train accuracy is high in the majority of the case and they both meet at one point.

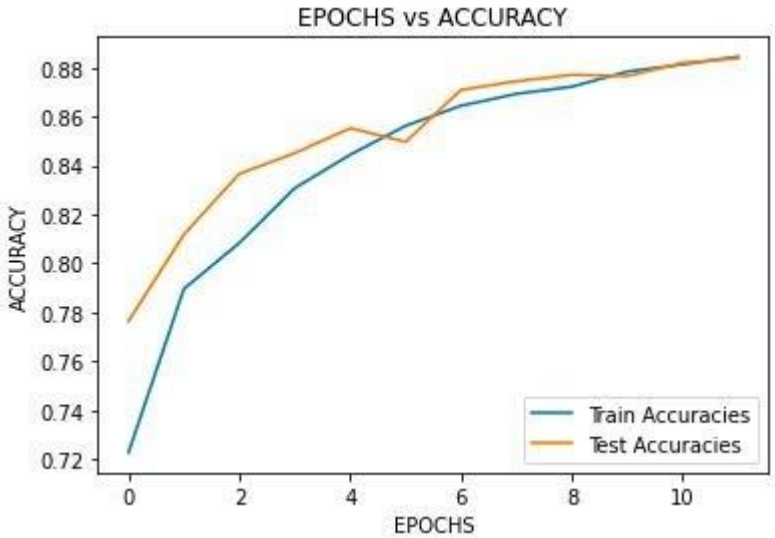
Setup :2



In this setup, Train accuracy is high in the majority of the case and they both meet at one point.

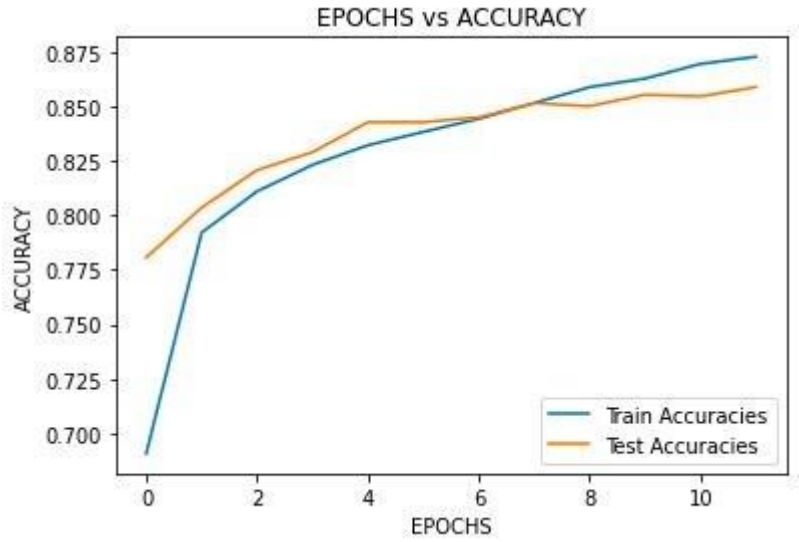
And with the increasing number of epochs training accuracy increases.

Setup :3



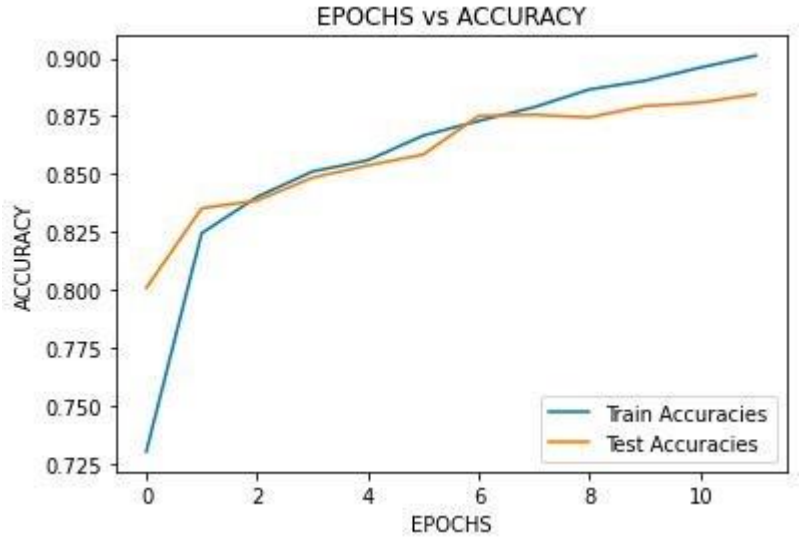
In this setup, Train accuracy is the same as testing accuracy. At the maximum number of epoch, they both have same accuracy.

Setup :4



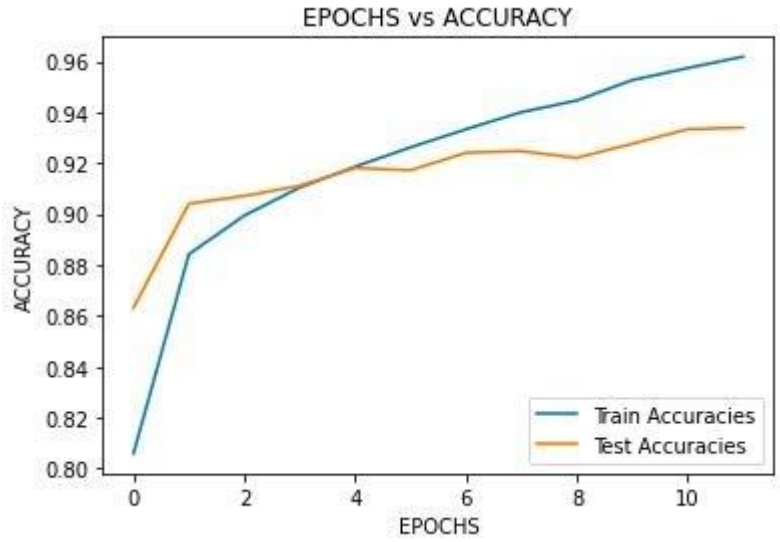
In this setup, Train accuracy is higher than testing accuracy. Initially when the number of epochs is less than 6, testing accuracy is greater than training accuracy. With the increasing number of epochs, graphs change.

Setup :5



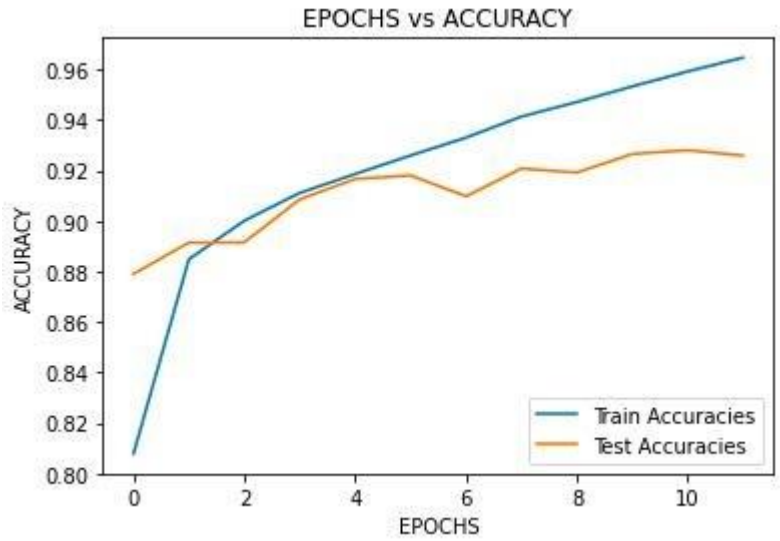
In this setup, Train accuracy is higher than testing accuracy. Initially when the number of epochs is less than 3, testing accuracy is greater than training accuracy. Both these meet at 2different points throughout the process.

Setup :6



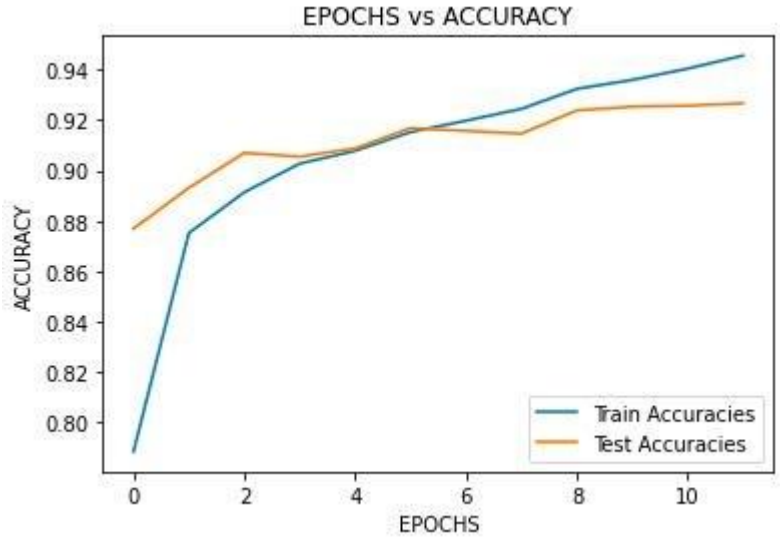
In this setup, Train accuracy is higher than testing accuracy. Initially when the number of epochs is less than 3, testing accuracy is greater than training accuracy. With the increasing number of epochs, graphs change. Both of these meet at one point.

Setup :7



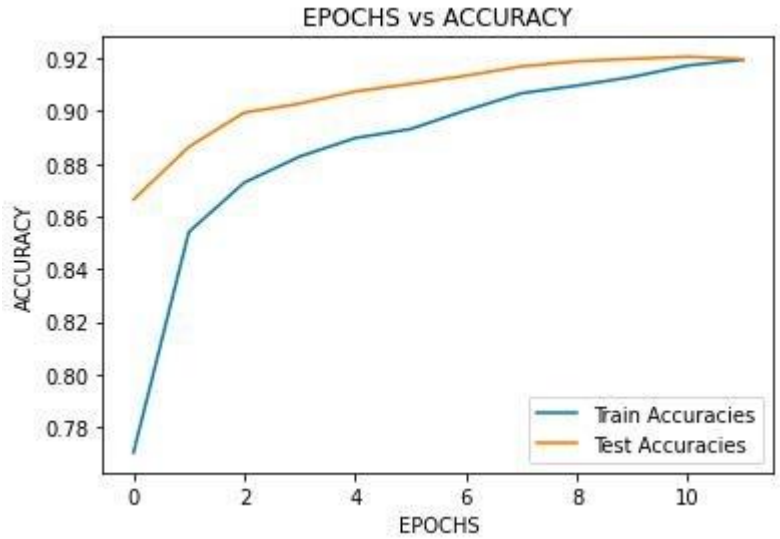
In this setup, Train accuracy is higher than testing accuracy.
They both meet at two points in this setup. Initially when the number of epochs is less than 6, testing accuracy is greater than training accuracy. With the increasing number of epochs, graphs change.

Setup :8



In this setup,testing accuracy is fluctuating irrespective of the number of epochs. Train accuracy is higher than testing accuracy. With the increasing number of epochs, graphs change and they both meet at one point.

Setup :9



In this setup,throughout the graph, test accuracy is higher than training accuracy.

3. Provide a detailed analysis and reasoning about the CNN setups that you tried.

There are 9 setups in total.
These are slight variations of original architecture, among all the setups the accuracy occurred when :
Droupt= 0.5

Optimizer=Adem
Stride=1
Kernel size=2

4. Discuss briefly methods that you used for data augmentation and why it helped to increase the accuracy of the model.

Here dataset size has been increased by four times(i.e from 60,000 to 300000) using the shifting Method.
Increasing dataset size will help the model to predict accurately, which helps to increase model accuracy .

References:

https://www.tutorialspoint.com/keras/keras_convolution_neural_network.htm

https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-th e-hyperparameters-in-a-deep-neural-network-d0604917584a

https://www.analyticsvidhya.com/blog/2021/11/training-neural-network-with-ker as-and-basics-of-deep-learning/

https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/#:~:text=The%20rectified%20linear%20activation%20function,otherwise%2C%20it%20will%20output%20zero.

https://keras.io/api/callbacks/early_stopping/

https://towardsdatascience.com/improving-accuracy-on-mnist-using-data-augmentation-b5c38eb5a903

https://www.kaggle.com/code/stefanie04736/simple-keras-model-with-k-fold-cross-validation/notebook