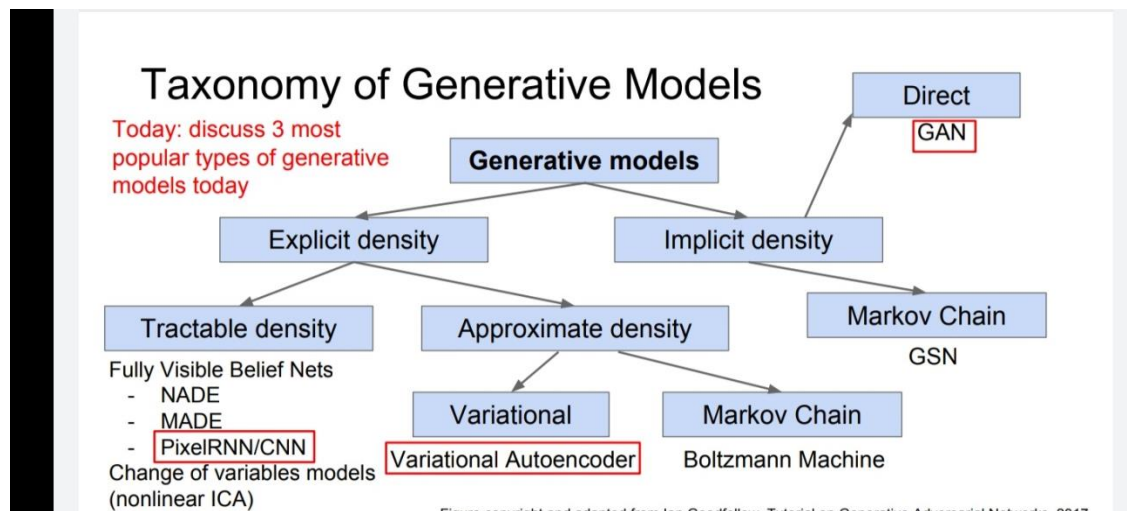


# GENERATIVE ADVERSARIAL NETWORKS

## OBJECTIVE:

This report mainly focuses on the Generative Adversarial Networks (GANs). GANs are one of the most rapidly growing areas of Computer Vision and it has a wide variety of research scope still left to work on. So, for a brief introduction to GANs. Here, is a flowchart to show the taxonomy of the Generative models.



Here, we have built a GANs network on the CIFAR-10 dataset and try to modify it in order to achieve more efficient and overall better performance of our network.

## TEST CASES:

### DATASET TRAINED:

I trained the model on a CIFAR-10 dataset. The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

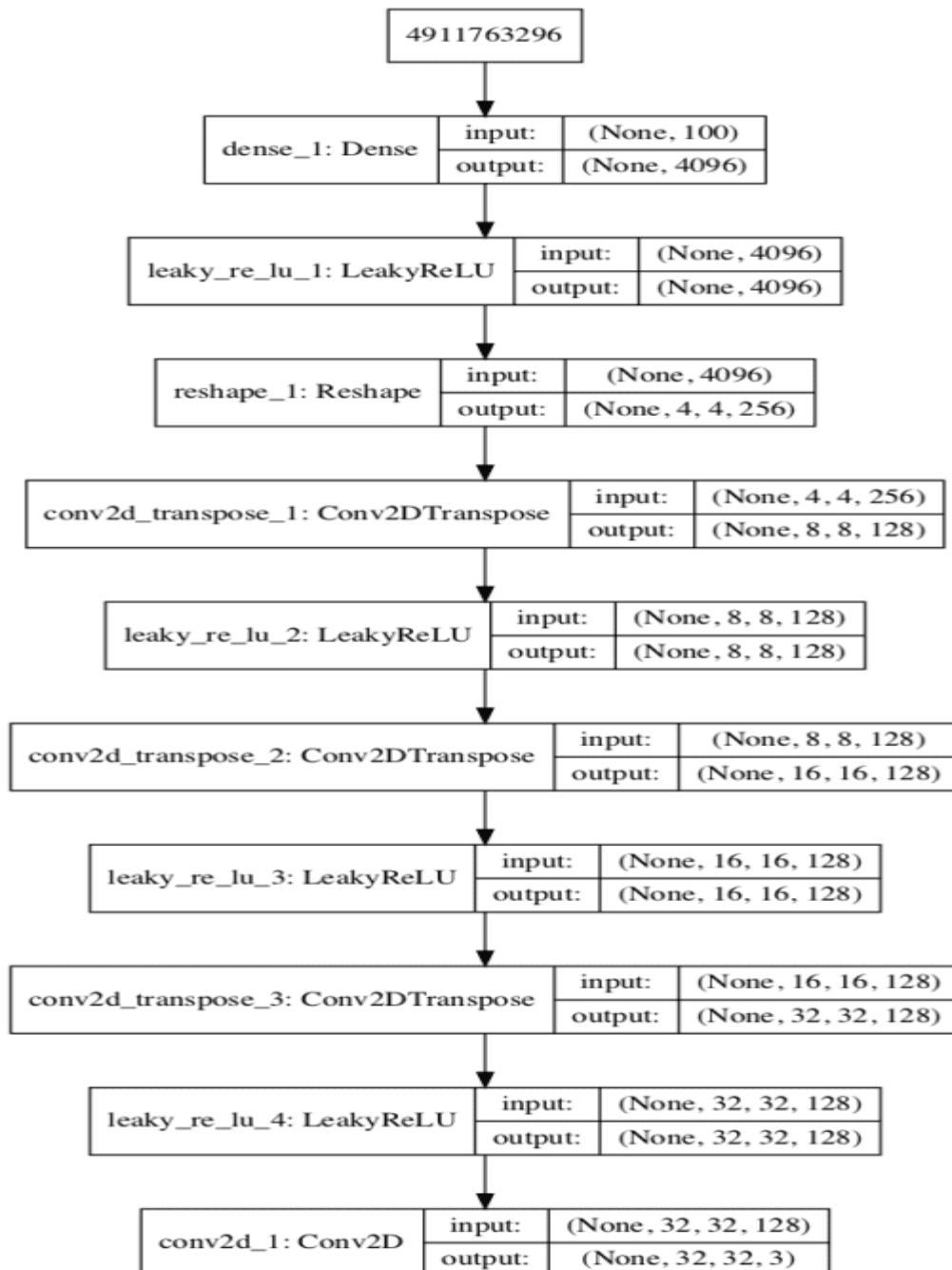
The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.

The classes range from Airplane, Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship and Truck.

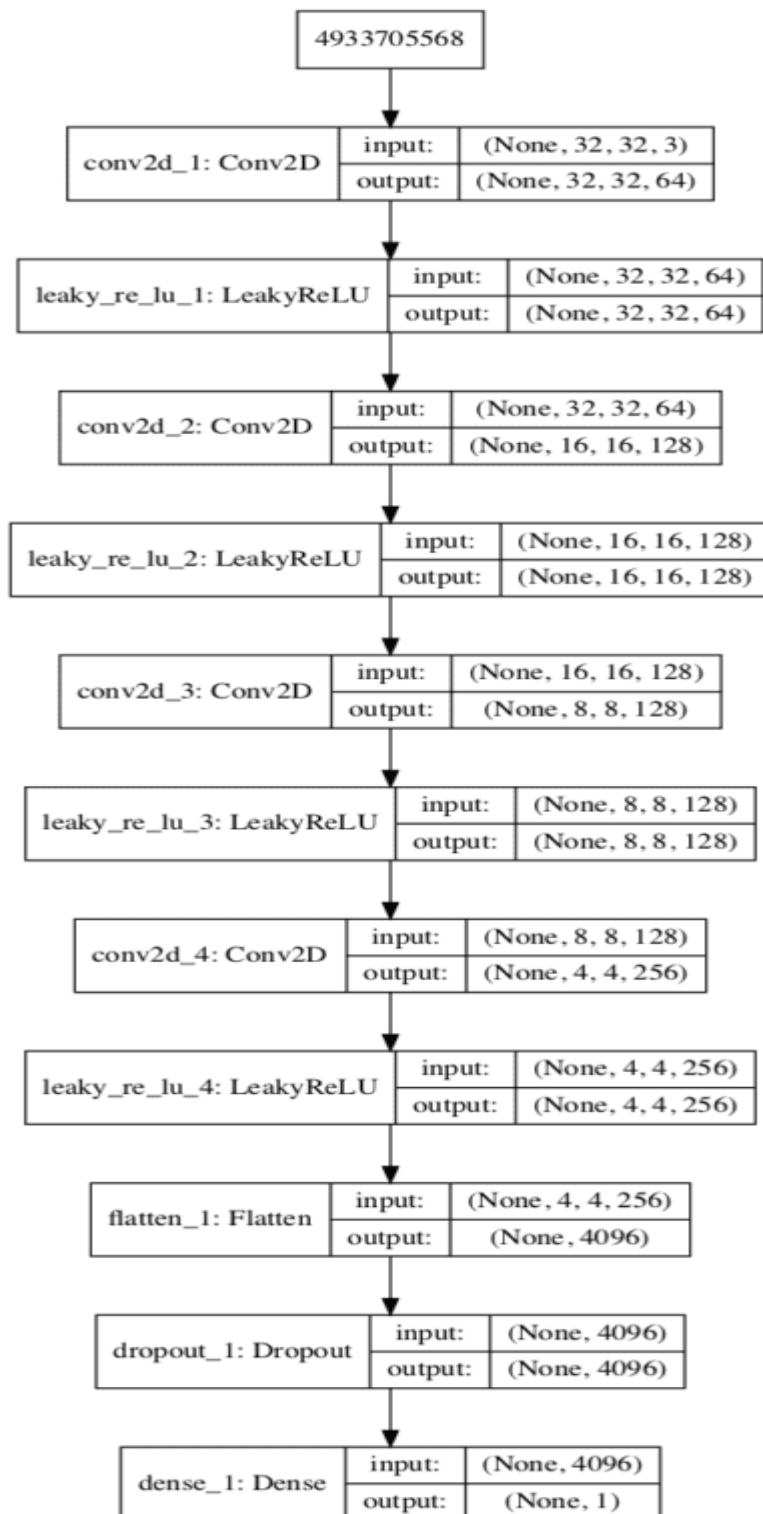
The classes are completely mutually exclusive. There is no overlap between automobiles and trucks. "Automobile" includes sedans, SUVs, things of that sort. "Truck" includes only big trucks. Neither includes pickup trucks.

### OUR MODEL:

Here, in brief the discriminator and generative model are shown.



This is our **generative model** with **three up sampling layers**.



This is our **discriminator model** with **three up sampling model**.

### **MODIFICATION IN THE MODEL:**

We have made some modification in the model which we are training. This modification includes

1. Introducing new layers in down sampling (generator part) and in up sampling part (discriminator part).
2. Introducing dropouts in the network
3. Changing the activation function from LeakyRelu to Relu or Sigmoid or any other function as listed below.
4. Changing the optimizer from Adam to RMSprop (performed best in the CIFAR-10 dataset image classification algorithm of optimizer comparison), Nadam, SGD, Adadelta (didn't performed better in image classification algorithm).

## PROBLEMS FACED:

The main problem faced was to decide the number of up sampling and down sampling.

## HOW TO OVERCOME THEM:

The number to select the amount of up sampling and down sampling was chosen on the basis of recommendation from the Stanford Lectures and some web crawling..

So, in both cases we chose **the number of layers to be 3**.

## COMPARISON:

Here are the table which are obtained after the modification.

Here, in the table,

D1 = loss by discriminator part while predicting the real image.

D2 = loss by discriminator part while predicting the fake image.

G = loss by the generative part.

### Introducing new layers in generative and discriminator model:

Initially, we have three down sampling layers in generative model while we correspondingly had three up sampling layers in discriminator model. Now, we will change this number to get a better model and will also record the data obtained after **epochs = 1 with batch-size = 10**.

Although epochs = 200 with batch-size = 128 is recommended but it takes around 3 hours to execute even on GPU mode. So, I reduced the numbers of both in order to see the response quick.

**Also the images generated after 200 epochs and batch-size = 128 is attached in later part**

No. of layers in both up sampling and down sampling	D1	D2	G	Accuracy	Fake

3	0.670	0.387	2.795	63	81
2	NOT GAVE SATISFACTORY RESULT..				
4					

### INTRODUCING DROPOUTS:

Dropouts are extensively used when there are lots of parameters and even when we lose some of the parameter then also other parameter can compensate the gone ones. Dropouts are widely used in Computer Vision areas. The popular ResNet model has dropout at each of its layer.

```

Layer (type)                Output Shape                Param #
=====
sequential_2 (Sequential)    (None, 32, 32, 3)          1466115
-----
sequential_1 (Sequential)    (None, 1)                   522497
=====
Total params: 1,988,612
Trainable params: 1,466,115
Non-trainable params: 522,497

```

The above shown data are form our model and here, we can see that we are having huge number of parameters especially in the discriminator part.

In our table we have used terms like Dropout = 0.5, which means to remove 50% of parameters from that particular layer.

In our model the **Discriminator model has Dropout = 0.4** initially. Now, table is the record of changing the dropout magnitude in the discriminator model.

Dropout	D1	D2	G	Accuracy	Fake
0.4	0.670	0.387	2.795	63	81
0.7	0.733	0.649	0.761	12	89
0.2	0.000	0.002	6.296	100	100

### CHANGING THE ACTIVATION FUNCTION:

Initially, we had LeakyRelu as the activation function on. Now, we are changing the activation function in our original model what type of changing in the losses we will obtain.

Activation Function	D1	D2	G	Accuracy	Fake
LeakyRelu	0.670	0.387	2.795	63	81
Relu	NOT GAVE SATISFACTORY REDULT.. IN SOME CASES THE MODEL FAILED TO RUN EVEN..				
ELU					
Sigmoid					

### CHANGING OPTIMIZER:

Initially, we had Adam as the optimizer but now, we will change them and will keep the hyperparameters values to the default to see the output.

Optimizer	D1	D2	G	Accuracy	Fake
Adam	0.670	0.387	2.795	63	81
RMSprop	0.701	0.687	0.705	5	100
Nadam	0.684	0.586	0.880	55	81
SGD	1.663	0.003	5.754	38	100
Adadelata	0.000	0.014	4.637	100	100

## YOUR INSIGHT:

Basically, GANs are two person game which both person work against each other to win the game. In the case Generative model and Destructive Model are the two person.

So, is the output result from the GANs used for this report.

This output are based on the default value of dropouts, hyper parameter, no. of epoch, batch size and all such stuffs.

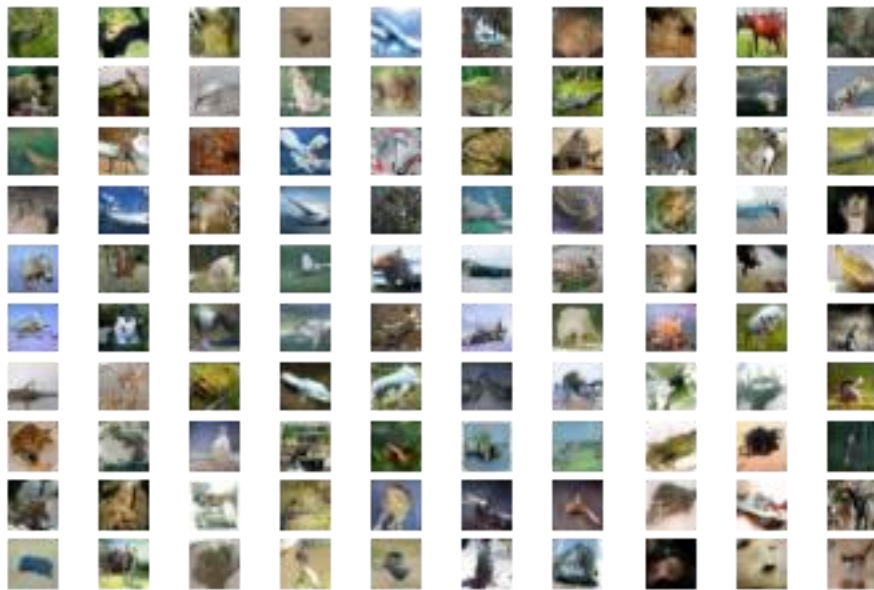


THIS WE GOT AFTER 10 EPOCHS..



THIS WE GOT AFTER 100 EPOCHS..

Finally, we got..



By seeing, the above result, we can say both the models are trying to fool each other. The generator part is trying to give better image to win over discriminator part while the discriminator part is trying to catch the treachery of the generator part..

## **FUTURE WORK:**

For this project, since it is a hot topic of research, the main area to think on is to design a better of both the generative and discriminator model. And, we might use an optimizer to learn the better latent value for generative part to generate better image and we can often work on better base model for the discriminator part to predict the treachery of generative part by past learning.