

# COMPUTER VISION TASK REPORT

Name: Shubham Rajnish Gandhi

ID: 2019A7PS0086G

First, I went through some introductory material on computer vision after which I read the articles on semi-supervised learning and its usefulness. I understood the theory more or less but the implementation part was where I struggled a lot. I chose to work with tensorflow as I felt more comfortable with it.

## Loading the dataset:

Initially, I used the [code](#) from the [STL-10](#) dataset site to load the dataset, which was rather long. It made sense to use tensorflow\_datasets to load the data which is what I did. I had a look at the dataset info and looked at the first few images and labels as well just to get a feel of what I was working with. The dataset had 5000 labelled train data, 8000 test data and 100000 unlabelled images as well with a shape of (96, 96, 3). There were 10 classes of images.

## Simple Supervised Learning Model:

I created a simple convolutional neural network using keras.Sequential. I performed rescaling by 1/255 because RGB data is in the range of 0 to 255 and such models work better with values between 0 and 1. I had three sets of alternating Conv2D (relu activation) and MaxPooling2D layers followed by a Flatten layer, a dense layer with 64 neurons and a final output dense layer with 10 neurons corresponding to 10 output classes with softmax activation.

I compiled the model with adam optimizer, loss as sparse categorical crossentropy, which is ideal for such labels that are NOT in one hot encoded format and chose accuracy as my metric.

I made an 80:20 train-val split and trained the model for 10 epochs. On evaluating the model on the test data, the accuracy hovered at around 52-53%.

The maximum val accuracy was around 56-57%. The rescaling actually played a crucial part without which the accuracy went as low as 30%.

### Trying ResNet50:

I also happened to try out the pretrained ResNet50 model but could not come up with impressive results. I performed the preprocessing that was expected by ResNet50 and used the pretrained imagenet weights. I used the same compile parameters as before. After training for 7 epochs, the accuracy on the test set came out to be around 38% and the maximum val accuracy was also around 38%.

### Further Tasks:

I went through the semi-supervised learning and self-supervised learning material and understood most of the theoretical concepts but could not implement pseudo labelling and SimCLR successfully.

A few cases where the semi-supervised learning approach might fail are as follows:

- If your labelled data size is significantly smaller than your unlabelled dataset, your models may tend to underfit, due to which a lot of your pseudo labels might be incorrect which would subsequently result in a large portion of your final training data to be incorrect thus producing poor outputs. A way to solve this might be image data augmentation wherein we perform different transformations on data to increase data size.
- Incorporating all of the unlabelled data into train would prove to be fatal because there is a high possibility that a lot of it is incorrectly labelled, instead it should be incorporated selectively.

Apart from achieving a high test set accuracy, there are a lot of metrics that should be considered when comparing models. These include the training time, computational resources used, overall efficiency, versatility and amount of fluctuation on changing parameters.