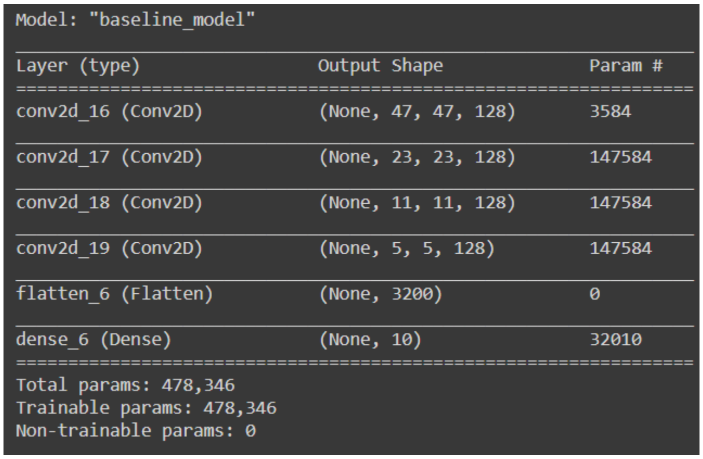
CV Task

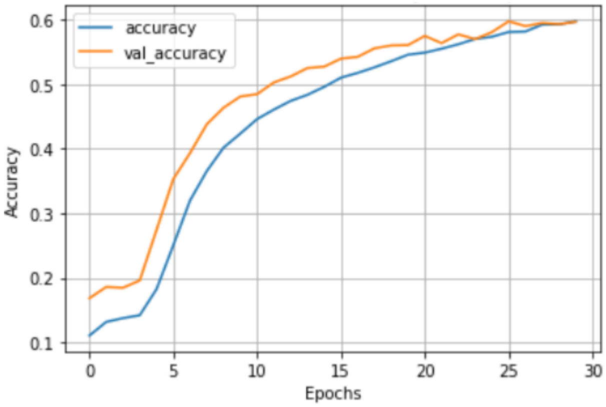
The STL10 dataset has been loaded from tensorflow_datasets which has 1,00,000 unlabeled data and 5,000 labeled data and 8,000 test data and each image has the dimensions of (96x96x3) with the number of classes as 10. After visualizing the images, checking labels I went forward with the tasks.

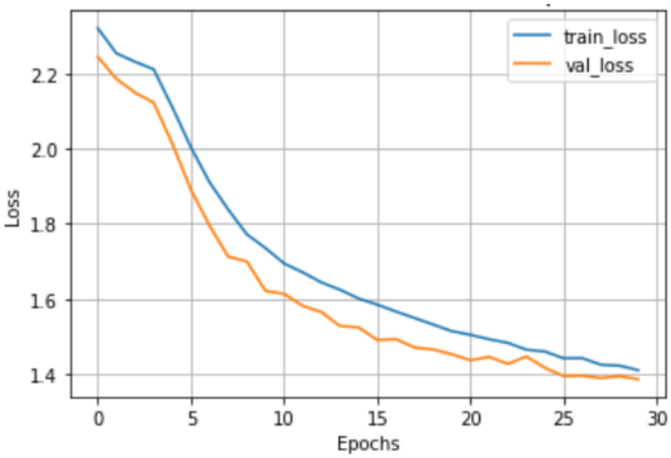
Supervised

1. Initially, I used a single convolutional layer that has a number of channels = 128, filter size = 3, stride = 2, relu activation and decided to work from there.
2. Next, I used 4 convolutional layers with the same parameters as (1).I have used weak image augmentations to avoid overfitting and I got a validation accuracy of 59
3. I also used ResNet-50 in the model architecture to experiment and got a validation accuracy of 68.7



```
Model: "baseline_model"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
conv2d_16 (Conv2D)          (None, 47, 47, 128)       3584

conv2d_17 (Conv2D)          (None, 23, 23, 128)       147584

conv2d_18 (Conv2D)          (None, 11, 11, 128)       147584

conv2d_19 (Conv2D)          (None, 5, 5, 128)         147584

flatten_6 (Flatten)         (None, 3200)              0

dense_6 (Dense)             (None, 10)                32010

=================================================================
Total params: 478,346
Trainable params: 478,346
Non-trainable params: 0
```





Semi-Supervised

1. For this task, I used the pseudo-labeling method. In this technique instead of giving labels to the unlabelled data, we give it approximate labels on the basis of labeled data.
2. Labeled data is first trained on the same baseline model we use for supervised learning
3. We then predict our pseudo labels by evaluating randomly selected unlabelled data by using model.predict() and we use a batch size of 128
4. Now, we want the predictions of classes from the vector representation obtained from the previous steps(expressed in a softmax distribution) to be above a particular threshold(I have taken 0.6) iteratively and we take batches of 128 and once we get all the values above the threshold, we take the maximum out of them and associate/concatenate that pseudo label with the unlabelled data.
5. The labels that do not cross the threshold are taken as a default -1

6. Finally, we append these labels and images to lists and train on them with the same baseline model
7. Once the model has trained on the pseudo labels dataset, we can retrain by using labeled data by building a dense layer classifier on top of the model and we get a test accuracy of

	Supervised Validation Accuracy	Semi-Supervised Validation Accuracy	Change in Accuracy
CNN Model	59.54	64.62	5.08
ResNet-50 Model	68.73	72	3.27

Self-Supervised

1. For this task, I used the SimClr framework for contrastive learning. This basically learns representations for an image(anchor image) augmentations(positive image) and the distance between the feature vectors having similar features is minimized and the distance between feature vectors of the augmentations is any random negative image is maximized.
2. For getting feature representations I used a convolutional architecture encoder which is trained and the weights are frozen and pass it through a dense layer that has units as the number of classes.
3. The two most important image augmentations I have used are
4. Cropping: picks out different parts of the image for encoding. This is carried out using
    - RandomZoom and RandomTranslation layers
    - Color jitter: This distorts the color in the images
5. We train an encoder on unlabeled images with a contrastive loss and an MLP is added on top of the encoder to improve the feature representations.
6. We use N-pair loss which is interpreted in the following way:
    - We treat each image in the batch as if it had its own class and generate a pair of augmentations

- Then each representation from the encoder is compared to other image augmentations and similarity between the same feature vectors is decreased and between different feature, vectors are increased.
- We use temperature-scaled cosine similarity.

7. The metrics used are validation accuracy and contrastive accuracy which is the ratio of cases in which the representation of an image is more similar to its augmentations as compared to different images. We get a validation accuracy of 53.30%

When do you expect semi-supervised learning approaches to fail? What can you do to avoid this?

1. Semi-supervised learning approaches fail when the initial labeled data are not sufficient to determine the underlying structure of data usually seen in clusters. The algorithm collapses when it is not able to assign different clusters and all images of different labels are assigned to a single cluster.
2. When the continuity assumption of points in close proximity are likely to share the same label is broken due to the presence of many outliers or few labeled data points which assigns wrong pseudo labels
3. Another issue arises when the initial labeled data doesn't include some classes which are then merged with other clusters. This leads to mislabeling of the pseudo labels
4. While predicting the pseudo labels, we assume that the model confidence increases over time. But, this need not be the case as the model may wrongly predict labels as well. In this case, there will be a bad feedback loop and semi-supervised learning may end up deteriorating the performance.

Apart from achieving a high test set accuracy, what other metrics do you think are important while comparing and contrasting different learning approaches?

1. The metrics used largely depend on the kind of task at hand. For example, if we are working on an application of ML in security(Say face detection) then, prioritizing to reduce false positives is much more crucial as it invades the system and can lead to mishappenings. False negatives can be dealt with by the person changing the angle of his

face so on and eventually he will get through. Having a metric that never allows false positives is crucial in this case.

2. The confusion matrix forms the basis for many other metrics such as AUC, F1 score.
3. The area under curve(AUC) is used for binary classification. AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example over a negative chosen example.
4. Another method used for test accuracy is F1 score which is the harmonic mean of precision and recall. Precision is the number of positive samples divided by the predicted positive samples. Recall is the number of correct positive results divided all samples that should be identified as positive.

BONUS:

A)Using AutoAugment

______I tried to implement semi-supervised tasks using SimClr and augment images using AutoAugment method. The operations we will be using are shearing, translating, rotation, auto_contrasting, brightness, sharpness, cutout, etc., and the policies for each augmentation are selected randomly and applied in our dataset for producing image augmentations.

The idea of AutoAugment is to learn the best data augmentation policies for a given dataset with the help of reinforcement learning by restricting it to choices specific to the dataset. For example, the best policies for CIFAR10 are colour based transformations. For SHVN, it is shearing the images. However, while dealing datasets of similar nature of objects, we can share the same data augmentations with them like we can use augmentation policies from CIFAR10 and apply to ImageNet or STL10, and similarly, augmentations of MNIST can be applied for SHVN. Data augmentation on CIFAR10 along a specific axis might be nonsensical for (say) medical images

B) Akin to how transfer learning is used in the supervised setting, can image representations learned in a self-supervised manner be transferred as well?

There are self-supervised ways of carrying out the transfer of image representations.

We first acquire the prior knowledge from the source domain by training with manual labels then we introduce target images and apply K-means upon the feature representations of the images to construct soft constraints between target samples and then this confidence is measured. Self-supervision is then introduced to make up for the ambiguous features by slightly augmenting the images and finding the relation between the images. Finally, by jointly trained with the prior knowledge and self-supervision the model learns a discriminative embedding space as wells as decision boundaries for target images.

C)How could one go about combining semi and self-supervised training to achieve good performance?

1) We can first create a feature-based clustering in embedding space using a partially labeled dataset and an unlabelled dataset. Then different images are related using a self-supervised contrastive learning approach by pulling augmented versions of the same images together and pushing different images away from the original image.
2) We can first train using self-supervised based contrastive learning and reduce the distance between images and their augmentations and increase the distance between the image and other images, then semi-supervised fine-tuning on the previous model/encoders and predicting pseudo-labels.