

Understanding Deep Learning Requires Rethinking Generalization

Team: Okay

- Madhav Agarwal (2020900022)
- Siddhant Bansal (2019900091)
- Piyush Singh (2020701031)

Github: <https://github.com/mdv3101/Okay>

Introduction



- Deep Neural Networks have the ability to generalise well on the given dataset.
- They show very small difference in the training and test performance.
- But the questions to consider here are:
 - What gives them the ability to generalize well?
 - Are they learning the 'actual' association?
 - Can we train the network to learn random associations?
 - Does regularization play a significant role while training a deep network?
- Due to the way in which neural networks are formulated, we cannot use traditional methods (VC dimension, uniform stability, etc.) to understand the generalization ability.

Introduction



- This work shows that the traditional view of generalization is *incapable of distinguishing between different neural networks that have different generalization performance*.
- For that purpose we do the following experiments:
 - **Randomized testing**
 - As the name suggests, here the neural networks are provided a different set of inputs and outputs and the performance is checked.
 - **Role played by *explicit* regularization**
 - Here we try to understand how well various regularization techniques like, weight decay, dropout, and data augmentation, are playing a role in generalization.
 - **Role played by *implicit* regularization**
 - Implicit regularization techniques such as batch norm, SGD and early stopping are applied to check the performance of neural network

Randomized Testing



- Here the goal is to see how well the neural networks generalize to data which is not 'logical' for us humans.
- The neural network is trained with original labels (baseline) and then it is trained using *random* labels.
- Training with *random* labels makes sure that there is no 'relationship' between the examples and the labels.
- What should we expect in this case?
 - Logically, this should make learning impossible.
 - And, due to this, the training should not converge.
- But, the observations made are interesting!
- The training is NOT affected by this change and the model converges.

Randomized Testing



- To make sure that the results are what is being observed, the authors perform these set of experiments:
 - Using **true labels** as a baseline.
 - **Corrupting** the **labels** by replacing a label with a random class's label.
 - **Shuffling pixels** of the input image.
 - **Randomizing** the **pixels** of the input image.
 - Creating **noisy images** using Gaussian noise while using the variance and mean of the dataset.
- Logically, it feels like apart from the first case, the models should not converge but, as we will see in the slides ahead, the results are surprising.

Randomized Testing



- The following trends are observed in the graphs in the original paper
 - With zero input and zero output corruption, the training converges in very few number of iterations
 - The time to converge for different corruption scenarios has the order:
No corruption < Pixel shuffling < Gaussian pixels < Random pixels < Random labels
 - For label corruption experiment, the time to converge increases with increase in the percentage of corrupted labels
 - For label corruption experiment, the test error or generalization error increases with increase in the percentage of corrupted label
- In all randomization experiments, the model is able to converge ie achieve perfect training accuracy with sufficient number of iterations.

Randomized Testing

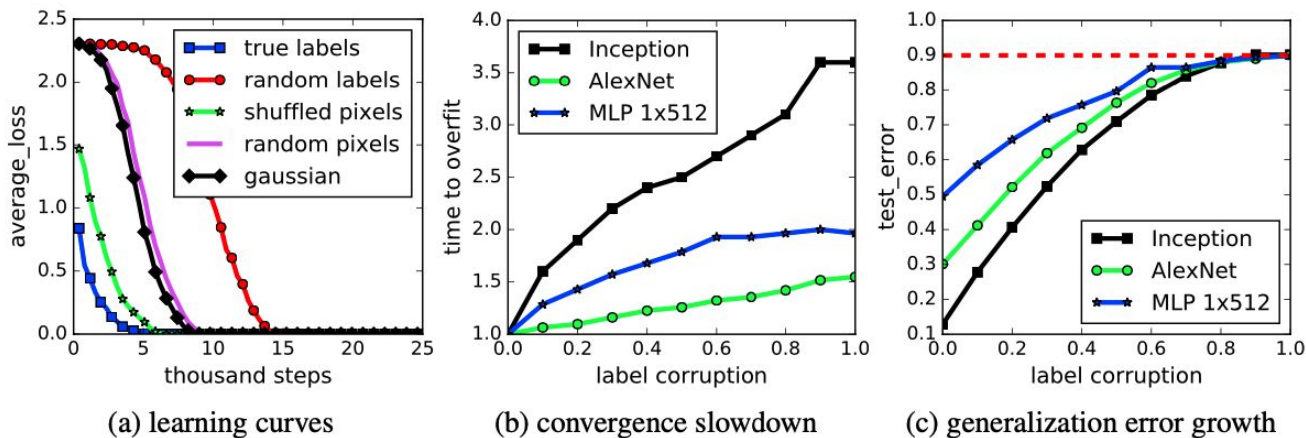


Figure 1: Fitting random labels and random pixels on CIFAR10. (a) shows the training loss of various experiment settings decaying with the training steps. (b) shows the relative convergence time with different label corruption ratio. (c) shows the test error (also the generalization error since training error is 0) under different label corruptions.

Role played by regularization



- We study the role of applying regularization on model training and its ability to generalize.
- There are some techniques like Bath Normalization which are not meant for regularization but indeed they gave regularization effect (Implicit Regularization).
- Other commonly used regularization techniques like weight decay, dropout and data augmentation are explicitly used to induce regularization (Explicit Regularization).
- The author argues that “Explicit regularization may improve generalization performance, but is neither necessary nor by itself sufficient for controlling generalization error.”
- The experiments suggests that tuning parameters help in improving generalization. The effect of regularization is minimal on generalization.

Role played by regularization

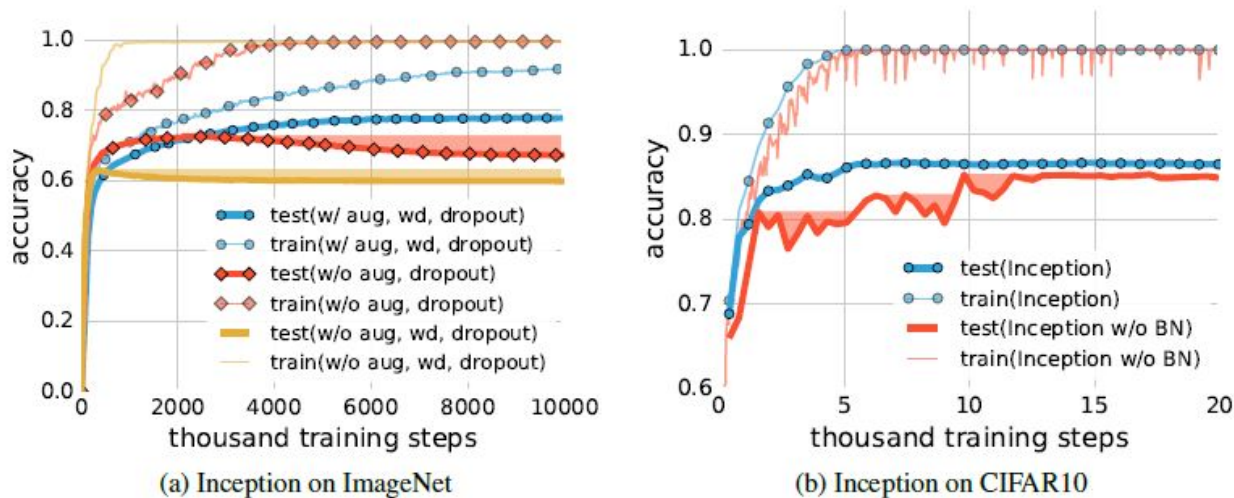


Figure 2: Effects of implicit regularizers on generalization performance. aug is data augmentation, wd is weight decay, BN is batch normalization. The shaded areas are the cumulative best test accuracy, as an indicator of potential performance gain of early stopping. (a) early stopping could potentially improve generalization when other regularizers are absent. (b) early stopping is not necessarily helpful on CIFAR10, but batch normalization stabilize the training process and improves generalization.

Datasets

CIFAR-10

- Dataset contains 50,000 training and 10,000 validation images
- Total Number of classes: 10
- Image size is 32x32 with 3 channels (cropped to 28x28)

airplane

automobile

bird

cat

deer

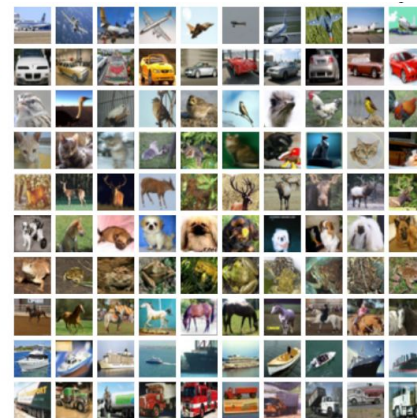
dog

frog

horse

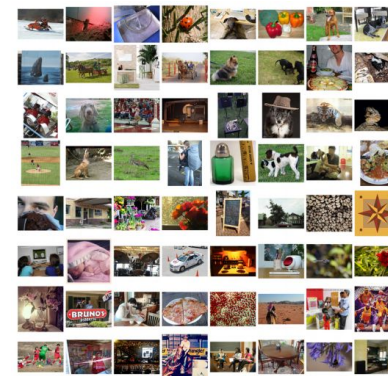
ship

truck



ImageNet

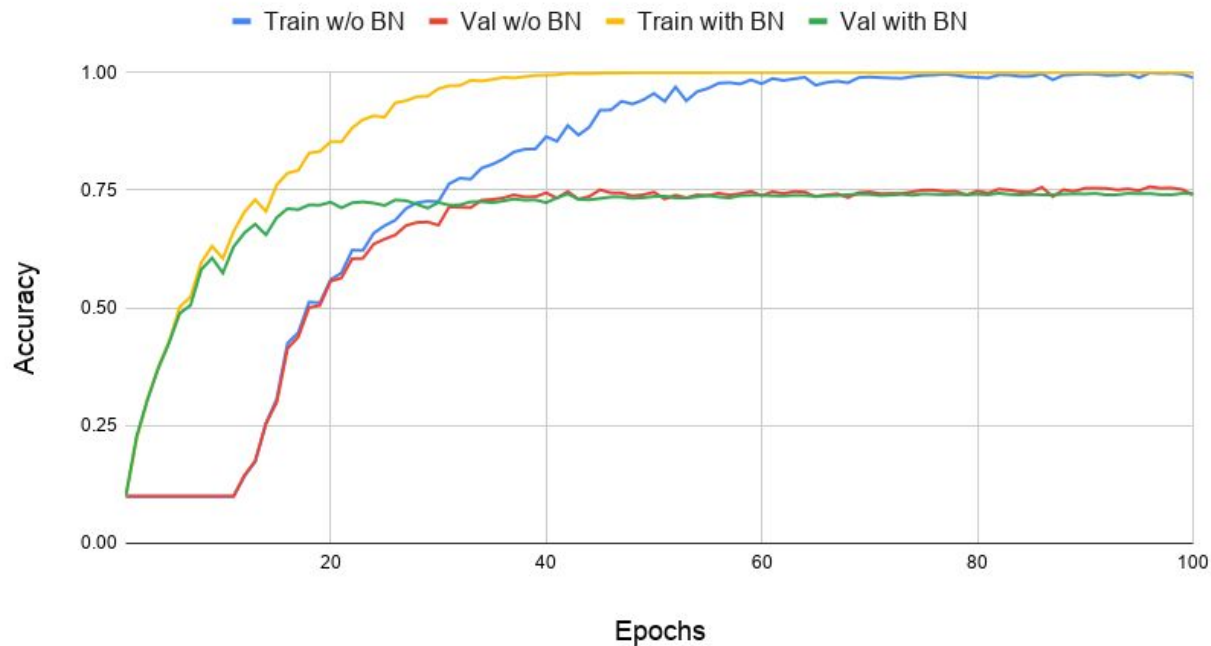
- Dataset contains 1,281,167 training and 50,000 validation images
- Total Number of classes: 1000
- Image resized to 299x299 with 3 channels



Our Results

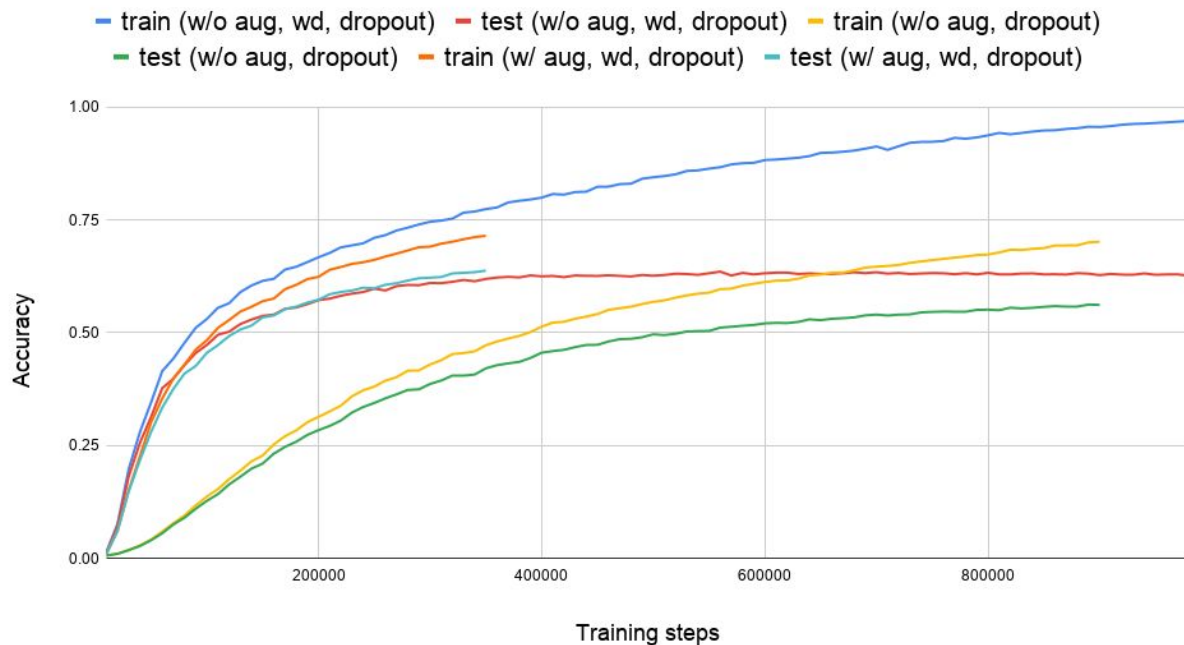
Our experiments on Regularization

Effect of Implicit Regularisation on Generalization Performance



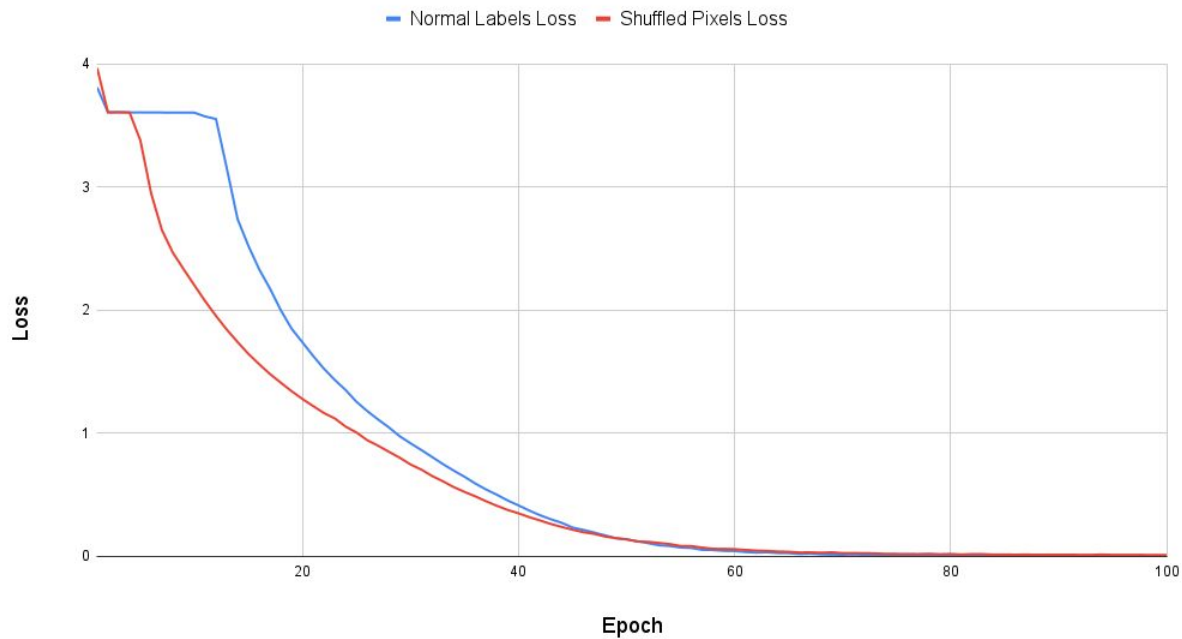
Our experiments on Regularization

Effect of Explicit Regularisation



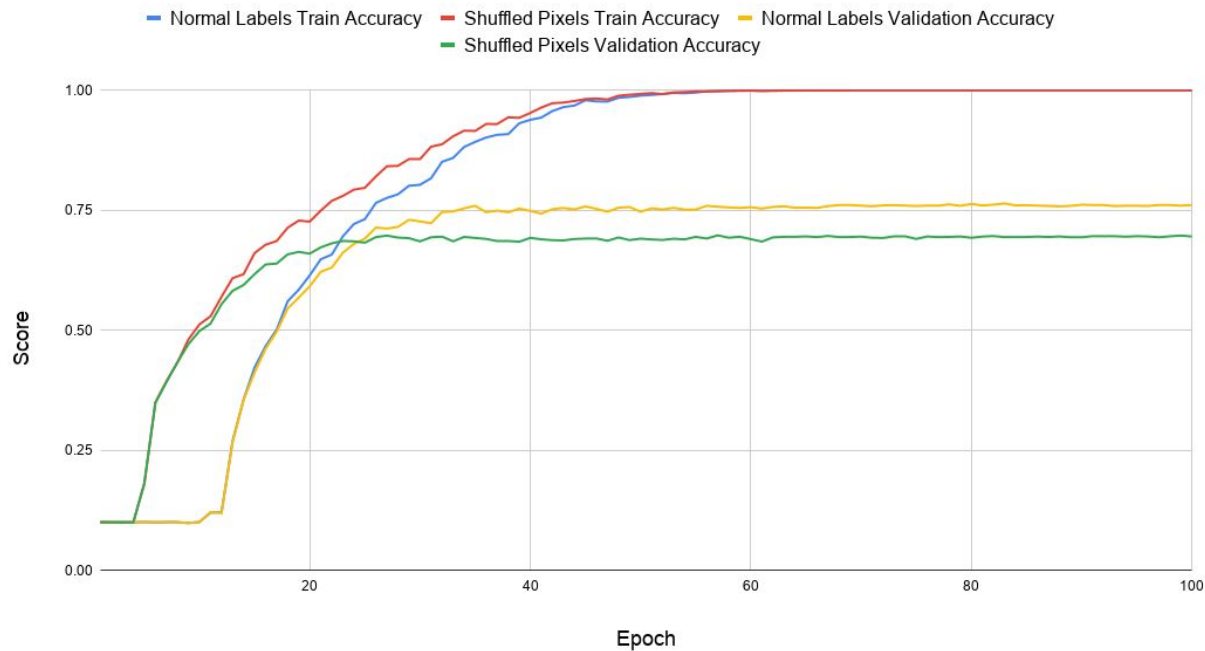
Effect of shuffling the pixels

Normal Inputs Loss vs. Shuffled Pixels Loss



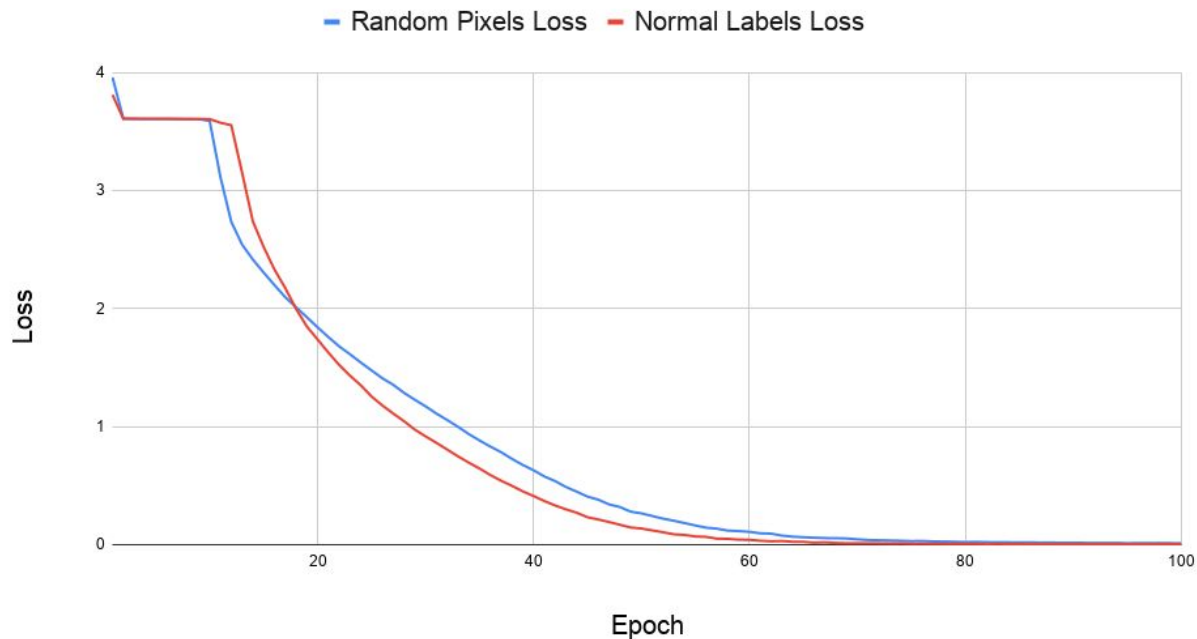
Effect of shuffling the pixels

Train and Validation accuracies for normal labels and shuffled pixels



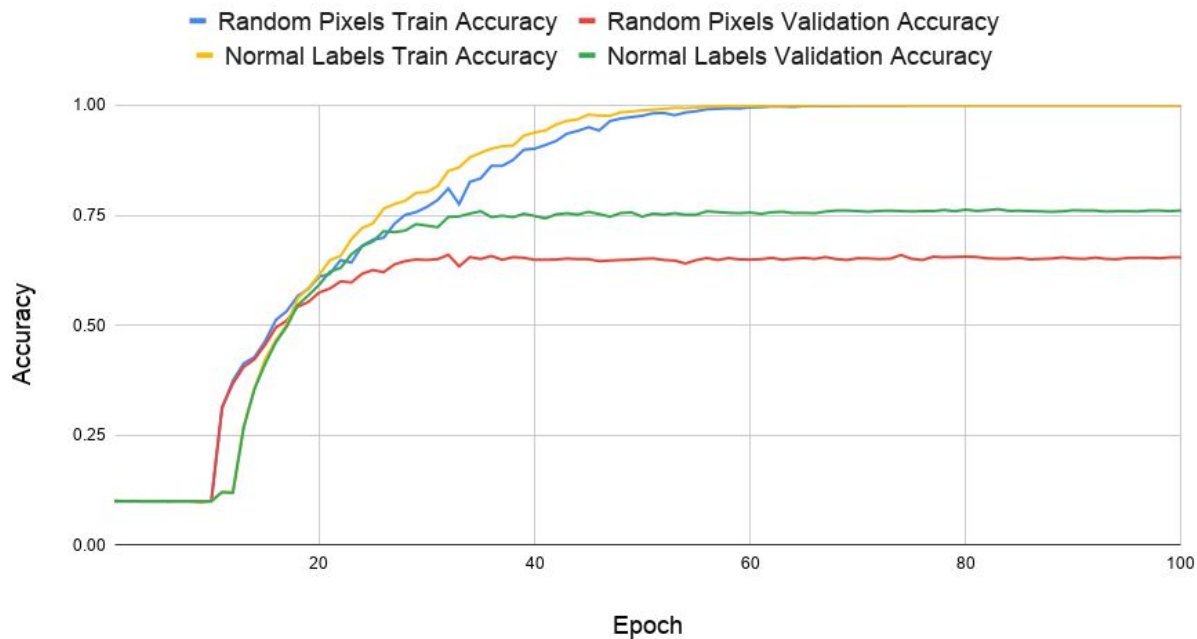
Effect of generating random input pixels

Normal Input Loss vs. Random Pixel Loss



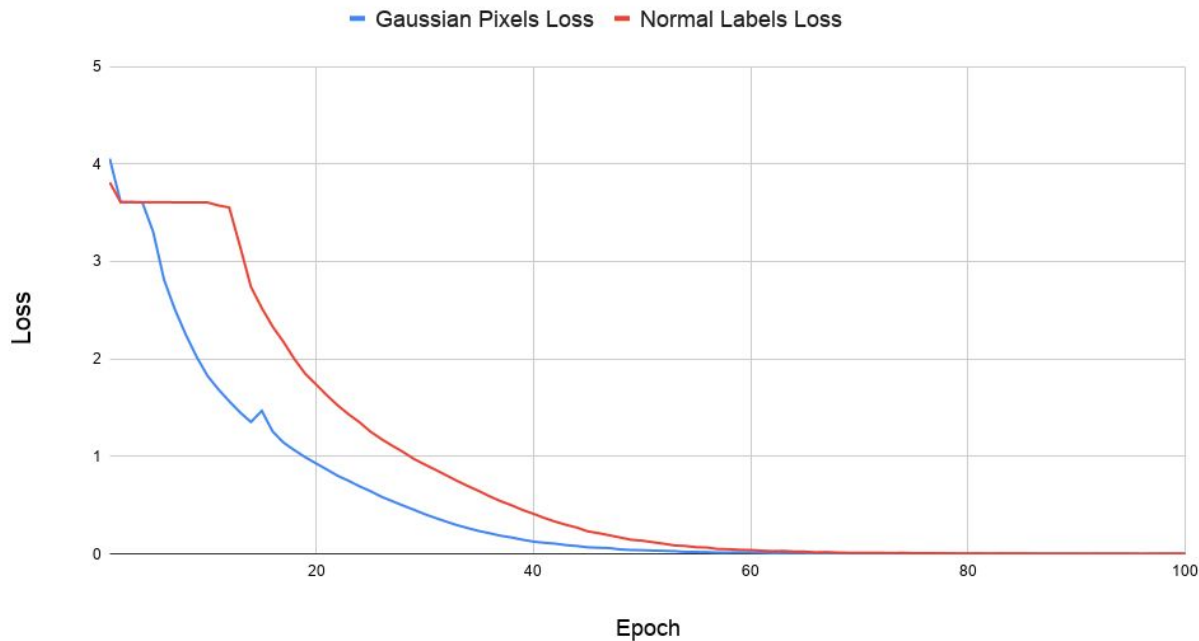
Effect of generating random input pixels

Train and Validation Accuracies for normal labels and random pixels



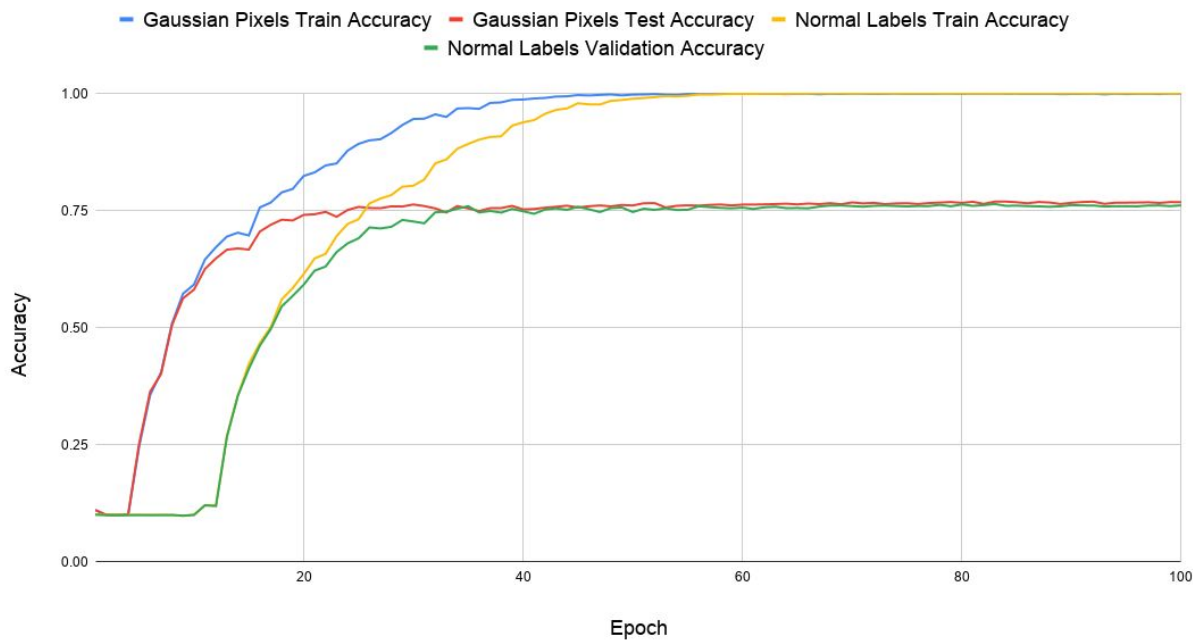
Effect of generating Gaussian input pixels

Normal Input Loss vs. Gaussian Input Loss



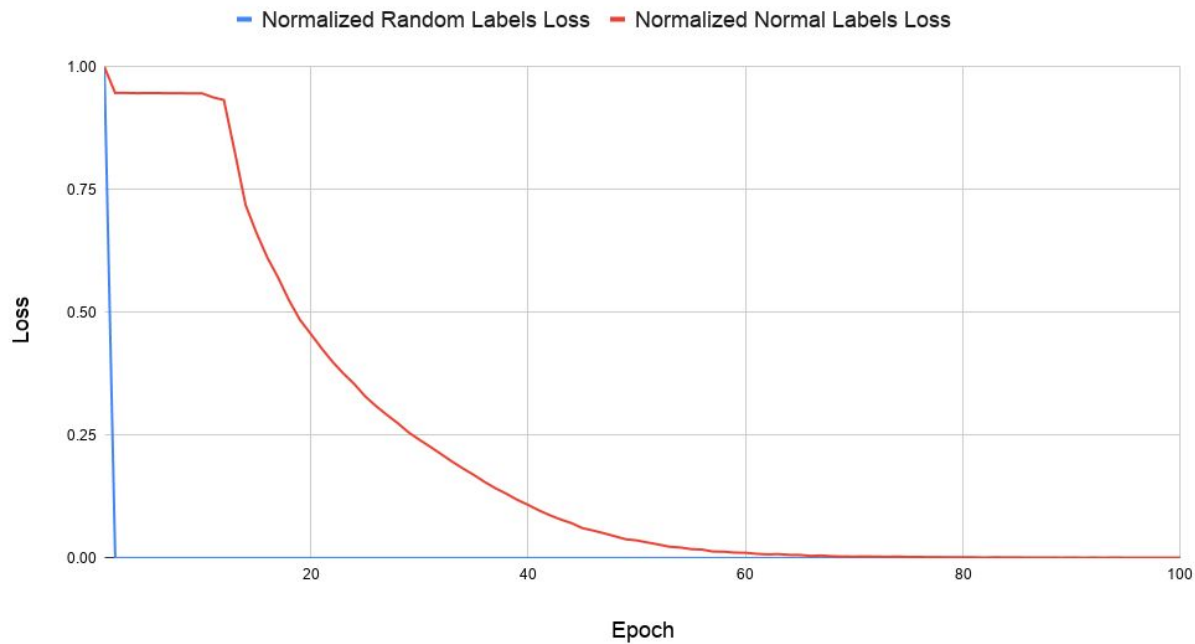
Effect of generating Gaussian input pixels

Effect of generating Gaussian input pixels



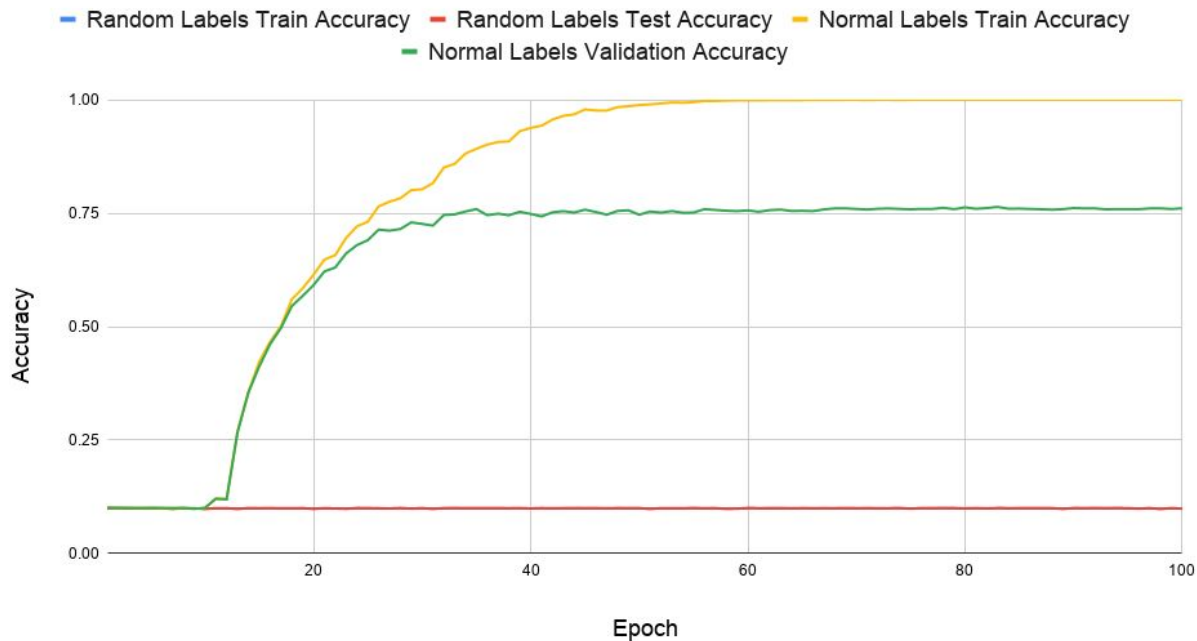
Effect of random labels

Random Input Loss vs. Normal Input Loss



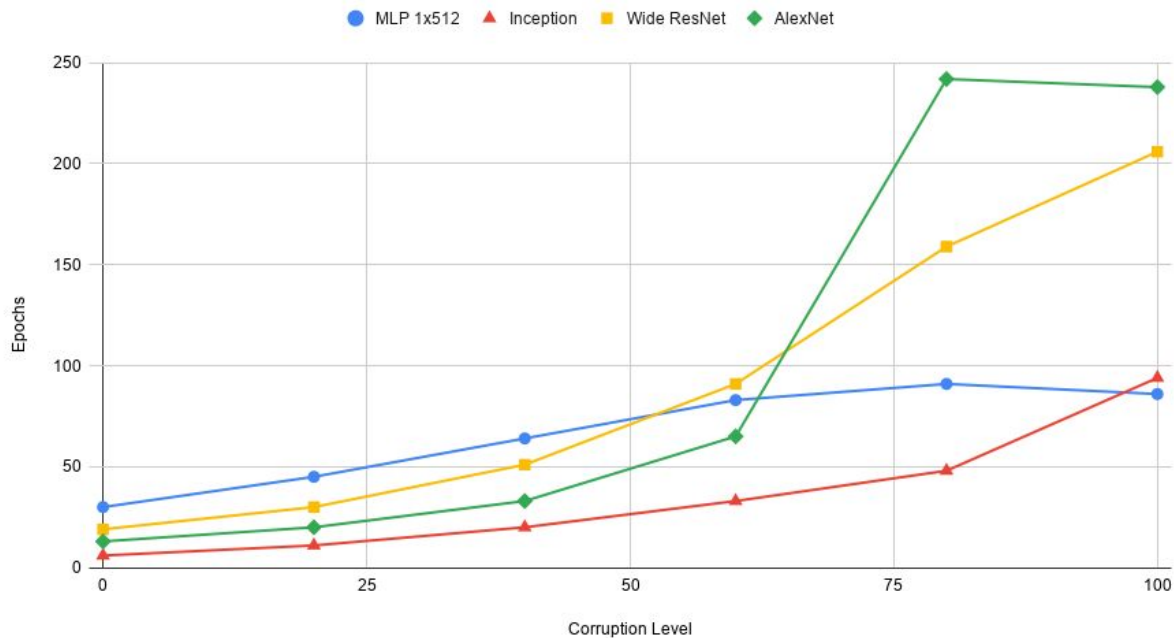
Effect of random labels

Effect of Random Labels



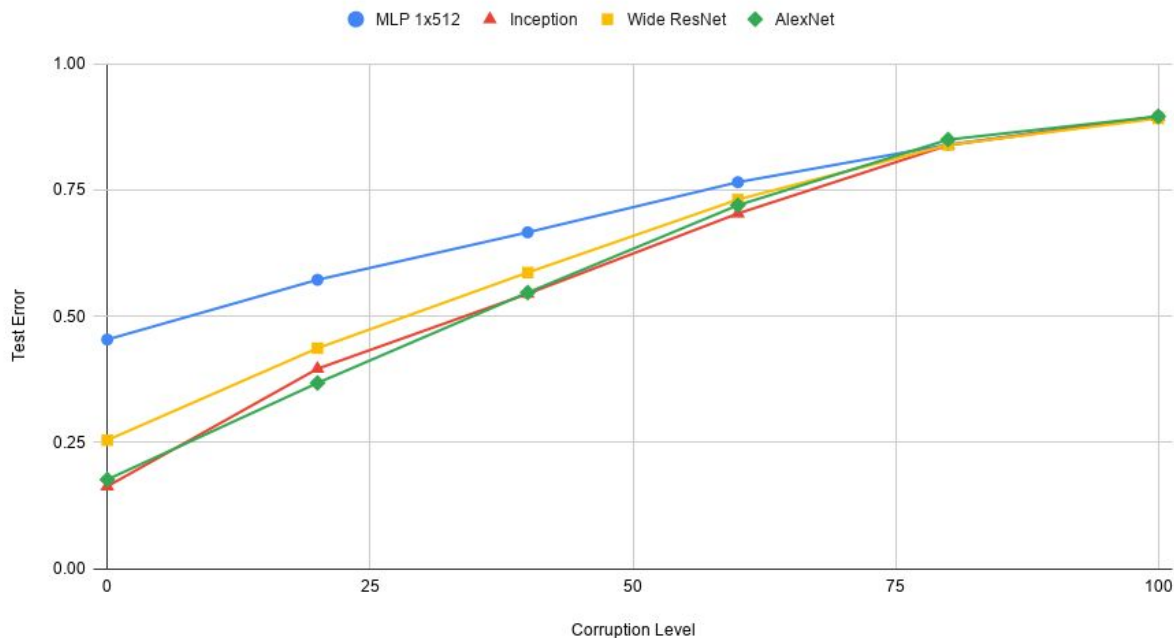
Label corruption: convergence

Epochs to converge vs label corruption level



Label corruption: Generalization Error

Generalization error vs label corruption level



Individual Contribution



Name	Task
Madhav Agarwal (2020900022)	Implemented the small Inception model, trained InceptionV3 on ImageNet, studied the effect of implicit regularization on CIFAR-10 dataset and, the effect of explicit regularization on the ImageNet dataset.
Siddhant Bansal (2019900091)	Studied the effect of data manipulation on the CIFAR-10 dataset using Inception (small), experiments on normal labels, random pixels, Gaussian noise, and random labels.
Piyush Singh (2020701031)	Studied the effect of label corruption ratio on model convergence and test error using AlexNet, MLP-512, Inception(small) and WideResnet on the CIFAR-10 dataset.